
django-fobi Documentation

Release 0.19.4

Artur Barseghyan <artur.barseghyan@gmail.com>

Aug 07, 2022

CONTENTS

1	Prerequisites	3
2	Key concepts	5
3	Main features and highlights	7
4	Roadmap	9
5	Some screenshots	11
6	Demo	13
6.1	Live demo	13
6.2	Run demo locally	13
7	Quick start	15
8	Installation	17
9	Creating a new form element plugin	21
9.1	Define and register the form element plugin	22
9.1.1	path/to/sample_textarea/fobi_form_elements.py	22
9.1.2	path/to/sample_textarea/forms.py	23
9.1.3	path/to/sample_textarea/widgets.py	24
9.1.4	path/to/sample_layout/fobi_form_elements.py	25
9.1.5	Form element plugin final steps	25
10	Creating a new form handler plugin	27
10.1	Define and register the form handler plugin	27
10.1.1	path/to/sample_mail/fobi_form_handlers.py	27
10.1.2	path/to/sample_mail/forms.py	28
10.1.3	Prioritise the execution order	29
10.1.4	Form handler plugin custom actions	29
10.1.5	Form handler plugin final steps	30
11	Creating a new form importer plugin	31
11.1	Define and register the form importer plugin	31
11.1.1	path/to/sample_importer/fobi_form_importers.py	31
11.1.2	path/to/sample_importer/forms.py	33
11.1.3	path/to/sample_importer/views.py	33
11.1.4	Form importer plugin final steps	35
12	Creating a form callback	37

13 Class-based views	39
13.1 Views	39
13.2 Permissions	39
14 Suggestions	41
14.1 Custom action for the form	41
14.2 When you want to customise too many things	41
15 Theming	43
15.1 Create a new theme	43
15.2 Make changes to an existing theme	47
15.2.1 Directory structure	47
15.2.2 fobi_themes.py	48
15.2.3 templates/override_simple_theme/base_view.html	48
15.2.4 templates/override_simple_theme/snippets/form_ajax.html	49
16 Form wizards	51
16.1 Basics	51
16.2 Bundled form wizard handler plugins	51
17 Integration with third-party apps and frameworks	53
17.1 Sample <i>IntegrationFormElementPlugin</i>	53
17.1.1 base.py	53
17.1.2 fobi_integration_form_elements.py	54
17.2 Sample <i>IntegrationFormHandlerPlugin</i>	55
17.2.1 base.py	55
17.2.2 fobi_integration_form_handlers.py	56
18 Permissions	57
19 Management commands	59
20 Tuning	61
21 Bundled plugins and themes	63
21.1 Bundled form element plugins	63
21.1.1 Fields	63
21.1.2 Content/presentation	64
21.1.3 Security	64
21.1.4 MPTT fields	64
21.1.5 Test	65
21.2 Bundled form handler plugins	65
21.3 Bundled themes	65
22 Third-party plugins and themes	67
23 HTML5 fields	69
24 Loading initial data using GET arguments	71
25 Dynamic initial values	73
26 Submitted form element plugins values	75
27 Rendering forms using third-party libraries	77
27.1 Using <i>django-crispy-forms</i>	77

27.2	Using <i>django-floppyforms</i>	77
28	Import/export forms	79
29	Translations	81
29.1	Available translations	81
29.2	Overriding translations	81
30	Debugging	83
31	Testing	85
31.1	Browser tests	85
31.1.1	Set up ChromeDriver	86
31.1.2	Set up Firefox 47	86
31.1.3	Set up headless Firefox	86
31.1.4	Setup PhantomJS	87
32	Writing documentation	89
33	Troubleshooting	91
34	Contributing	93
35	License	95
36	Support	97
37	Author	99
38	Screenshots	101
38.1	Bootstrap3 theme	101
38.1.1	Dashboard	101
38.1.2	Create a form	102
38.1.3	View/edit form	103
38.1.3.1	Form elements	103
38.1.3.2	Form handlers	107
38.2	Simple theme	118
38.2.1	View/edit form	118
39	(Sub)modules	123
39.1	fobi.contrib.apps.djangocms_integration	123
39.1.1	Prerequisites	123
39.1.2	Installation	123
39.1.2.1	Versions	123
39.1.2.2	your_project/settings.py	123
39.1.3	Information for developers	124
39.1.3.1	Templates for DjangoCMS	124
39.1.3.2	Template rendering	124
39.1.3.3	Using custom templates for rendering the widget	124
39.1.4	Usage	125
39.2	fobi.contrib.apps.drf_integration	125
39.2.1	Live demo	126
39.2.2	Supported fields	126
39.2.2.1	Content (presentational form elements)	126
39.2.2.2	Fields	126
39.2.3	Not (yet) supported fields	127

39.2.4	Implementation details	127
39.2.5	Installation	129
39.2.5.1	Versions	129
39.2.5.2	<code>your_project/settings.py</code>	130
39.2.5.3	<code>your_project/urls.py</code>	131
39.2.6	Usage	131
39.2.6.1	LIST	131
39.2.6.2	OPTIONS	131
39.2.6.3	PUT	136
39.2.7	Callbacks	136
39.2.7.1	<code>fobi_form_callbacks.py</code>	136
39.2.8	Testing	137
39.2.9	Limitations	137
39.3	Form elements	137
39.3.1	Content form element	137
39.3.1.1	<code>fobi.contrib.apps.drf_integration.form_elements.content.content_image</code>	137
39.3.1.1.1	Installation	137
39.3.1.1.2	Usage	138
39.3.1.2	<code>fobi.contrib.apps.drf_integration.form_elements.content.content_image_url</code>	138
39.3.1.2.1	Installation	138
39.3.1.2.2	Usage	139
39.3.1.3	<code>fobi.contrib.apps.drf_integration.form_elements.content.content_markdown</code>	140
39.3.1.3.1	Installation	140
39.3.1.3.2	Usage	140
39.3.1.4	<code>fobi.contrib.apps.drf_integration.form_elements.content.content_richtext</code>	141
39.3.1.4.1	Installation	141
39.3.1.4.2	Usage	141
39.3.1.5	<code>fobi.contrib.apps.drf_integration.form_elements.content.content_text</code>	142
39.3.1.5.1	Installation	142
39.3.1.5.2	Usage	142
39.3.1.6	<code>fobi.contrib.apps.drf_integration.form_elements.content.content_video</code>	143
39.3.1.6.1	Installation	143
39.3.1.6.2	Usage	143
39.3.2	Fields	144
39.3.2.1	<code>fobi.contrib.apps.drf_integration.form_elements.fields.boolean</code>	144
39.3.2.1.1	Installation	144
39.3.2.2	<code>fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple</code>	145
39.3.2.2.1	Installation	145
39.3.2.3	<code>fobi.contrib.apps.drf_integration.form_elements.fields.date</code>	145
39.3.2.3.1	Installation	145
39.3.2.4	<code>fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down</code>	146
39.3.2.4.1	Installation	146
39.3.2.5	<code>fobi.contrib.apps.drf_integration.form_elements.fields.datetime</code>	146
39.3.2.5.1	Installation	146
39.3.2.6	<code>fobi.contrib.apps.drf_integration.form_elements.fields.decimal</code>	147
39.3.2.6.1	Installation	147
39.3.2.7	<code>fobi.contrib.apps.drf_integration.form_elements.fields.duration</code>	147
39.3.2.7.1	Installation	147
39.3.2.8	<code>fobi.contrib.apps.drf_integration.form_elements.fields.email</code>	148
39.3.2.8.1	Installation	148
39.3.2.9	<code>fobi.contrib.apps.drf_integration.form_elements.fields.file</code>	148
39.3.2.9.1	Installation	148
39.3.2.10	<code>fobi.contrib.apps.drf_integration.form_elements.fields.float</code>	149
39.3.2.10.1	Installation	149

39.3.2.11	fobi.contrib.apps.drf_integration.form_elements.fields.hidden	149
39.3.2.11.1	Installation	149
39.3.2.12	fobi.contrib.apps.drf_integration.form_elements.fields.input	150
39.3.2.12.1	Installation	150
39.3.2.13	fobi.contrib.apps.drf_integration.form_elements.fields.integer	150
39.3.2.13.1	Installation	150
39.3.2.14	fobi.contrib.apps.drf_integration.form_elements.fields.ip_address	151
39.3.2.14.1	Installation	151
39.3.2.15	fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean	151
39.3.2.15.1	Installation	151
39.3.2.16	fobi.contrib.apps.drf_integration.form_elements.fields.password	152
39.3.2.16.1	Installation	152
39.3.2.17	fobi.contrib.apps.drf_integration.form_elements.fields.radio	152
39.3.2.17.1	Installation	152
39.3.2.18	fobi.contrib.apps.drf_integration.form_elements.fields.range_select	153
39.3.2.18.1	Installation	153
39.3.2.19	fobi.contrib.apps.drf_integration.form_elements.fields.regex	153
39.3.2.19.1	Installation	153
39.3.2.20	fobi.contrib.apps.drf_integration.form_elements.fields.select	154
39.3.2.20.1	Installation	154
39.3.2.21	fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object	154
39.3.2.21.1	Installation	154
39.3.2.22	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple	155
39.3.2.22.1	Installation	155
39.3.2.23	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects	155
39.3.2.23.1	Installation	155
39.3.2.24	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple	156
39.3.2.24.1	Installation	156
39.3.2.25	fobi.contrib.apps.drf_integration.form_elements.fields.slider	156
39.3.2.25.1	Installation	156
39.3.2.26	fobi.contrib.apps.drf_integration.form_elements.fields.slug	157
39.3.2.26.1	Installation	157
39.3.2.27	fobi.contrib.apps.drf_integration.form_elements.fields.text	157
39.3.2.27.1	Installation	157
39.3.2.28	fobi.contrib.apps.drf_integration.form_elements.fields.textarea	158
39.3.2.28.1	Installation	158
39.3.2.29	fobi.contrib.apps.drf_integration.form_elements.fields.time	158
39.3.2.29.1	Installation	158
39.3.2.30	fobi.contrib.apps.drf_integration.form_elements.fields.url	159
39.3.2.30.1	Installation	159
39.4	Form handlers	159
39.4.1	fobi.contrib.apps.drf_integration.form_handlers.db_store	159
39.4.1.1	Installation	159
39.4.2	fobi.contrib.apps.drf_integration.form_handlers.http_repost	160
39.4.2.1	Installation	160
39.4.3	fobi.contrib.apps.drf_integration.form_handlers.mail	160
39.4.3.1	Installation	160
39.4.4	fobi.contrib.apps.drf_integration.form_handlers.mail_sender	161
39.4.4.1	Installation	161
39.5	fobi.contrib.apps.feincms_integration	161
39.5.1	Prerequisites	161
39.5.2	Installation	161
39.5.2.1	Versions	161
39.5.2.2	your_project/settings.py	162

39.5.2.3	your_project/page/models.py	162
39.5.2.4	your_project/admin.py	163
39.5.2.5	Information for developers	163
39.5.2.5.1	Template rendering	163
39.5.2.6	Using custom templates for rendering the widget	163
39.5.3	Usage	164
39.6	fobi.contrib.apps.mezzanine_integration	164
39.6.1	Prerequisites	165
39.6.2	Installation	165
39.6.2.1	Versions	165
39.6.2.2	your_project/settings.py	165
39.6.3	Information for developers	165
39.6.3.1	Template rendering	165
39.6.3.2	Using custom templates for rendering the form	165
39.6.4	Usage	166
39.6.4.1	The fobi page model	166
39.6.4.2	Steps described	167
39.7	fobi.contrib.apps.wagtail_integration	167
39.7.1	Prerequisites	167
39.7.2	Installation	167
39.7.2.1	Versions	167
39.7.2.2	your_project/settings.py	167
39.7.2.3	your_project/your_app/models.py	167
39.7.2.4	Information for developers	168
39.7.2.4.1	Template rendering	168
39.7.2.5	Using custom templates for rendering the widget	168
39.7.3	Usage	169
40	Form elements	171
40.1	fobi.contrib.plugins.form_elements.content.content_image	171
40.1.1	Installation	171
40.2	fobi.contrib.plugins.form_elements.content.content_image_url	171
40.2.1	Installation	172
40.3	fobi.contrib.plugins.form_elements.content.content_markdown	172
40.3.1	Installation	172
40.4	fobi.contrib.plugins.form_elements.content.content_richtext	173
40.4.1	Installation	173
40.4.2	Controlling HTML tags and attributes	174
40.5	fobi.contrib.plugins.form_elements.content.content_text	174
40.5.1	Installation	174
40.6	fobi.contrib.plugins.form_elements.content.content_video	175
40.6.1	Installation	175
40.7	fobi.contrib.plugins.form_elements.fields.boolean	176
40.7.1	Installation	176
40.8	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple	176
40.8.1	Installation	176
40.8.2	Usage	177
40.9	fobi.contrib.plugins.form_elements.fields.date	178
40.9.1	Installation	178
40.10	fobi.contrib.plugins.form_elements.fields.date_drop_down	179
40.10.1	Installation	179
40.11	fobi.contrib.plugins.form_elements.fields.datetime	179
40.11.1	Installation	179
40.12	fobi.contrib.plugins.form_elements.fields.decimal	180

40.12.1	Installation	180
40.13	fobi.contrib.plugins.form_elements.fields.duration	180
40.13.1	Installation	180
40.13.2	Usage	181
40.14	fobi.contrib.plugins.form_elements.fields.email	181
40.14.1	Installation	181
40.15	fobi.contrib.plugins.form_elements.fields.file	182
40.15.1	Installation	182
40.16	fobi.contrib.plugins.form_elements.fields.float	182
40.16.1	Installation	182
40.17	fobi.contrib.plugins.form_elements.fields.hidden	183
40.17.1	Installation	183
40.18	fobi.contrib.plugins.form_elements.fields.input	183
40.18.1	Installation	184
40.19	fobi.contrib.plugins.form_elements.fields.integer	184
40.19.1	Installation	184
40.20	fobi.contrib.plugins.form_elements.fields.ip_address	184
40.20.1	Installation	185
40.21	fobi.contrib.plugins.form_elements.fields.null_boolean	185
40.21.1	Installation	185
40.22	fobi.contrib.plugins.form_elements.fields.password	185
40.22.1	Installation	186
40.23	fobi.contrib.plugins.form_elements.fields.radio	186
40.23.1	Installation	186
40.23.2	Usage	187
40.24	fobi.contrib.plugins.form_elements.fields.range_select	187
40.24.1	Installation	187
40.25	fobi.contrib.plugins.form_elements.fields.regex	189
40.25.1	Installation	189
40.26	fobi.contrib.plugins.form_elements.fields.select	189
40.26.1	Installation	189
40.26.2	Usage	190
40.27	fobi.contrib.plugins.form_elements.fields.select_model_object	190
40.27.1	Installation	190
40.28	fobi.contrib.plugins.form_elements.fields.select_mptt_model_object	191
40.28.1	Installation	191
40.28.1.1	Install <i>django-mptt</i>	191
40.28.1.2	Install <i>select_mptt_model_object</i> plugin	192
40.29	fobi.contrib.plugins.form_elements.fields.select_multiple	193
40.29.1	Installation	193
40.29.2	Usage	193
40.30	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects	194
40.30.1	Installation	194
40.31	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects	195
40.31.1	Installation	195
40.31.1.1	Install <i>django-mptt</i>	195
40.31.1.2	Install <i>select_multiple_mptt_model_objects</i> plugin	195
40.32	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max	196
40.32.1	Installation	196
40.32.2	Usage	197
40.33	fobi.contrib.plugins.form_elements.fields.slider	197
40.33.1	Installation	197
40.34	fobi.contrib.plugins.form_elements.fields.slug	198
40.34.1	Installation	198

40.35	fobi.contrib.plugins.form_elements.fields.text	199
40.35.1	Installation	199
40.36	fobi.contrib.plugins.form_elements.fields.textarea	199
40.36.1	Installation	199
40.37	fobi.contrib.plugins.form_elements.fields.time	200
40.37.1	Installation	200
40.38	fobi.contrib.plugins.form_elements.fields.url	200
40.38.1	Installation	200
40.39	fobi.contrib.plugins.form_elements.security.captcha	201
40.39.1	Prerequisites	201
40.39.2	Installation	201
40.39.2.1	Install <i>django-simple-captcha</i>	201
40.39.2.2	Install <i>fobi</i> Captcha plugin	201
40.39.3	Troubleshooting and usage limitations	202
40.39.3.1	In combination with other captcha solutions	202
40.39.4	Usage	202
40.40	fobi.contrib.plugins.form_elements.security.honeypot	202
40.40.1	Installation	202
40.41	fobi.contrib.plugins.form_elements.security.invisible_recaptcha	202
40.41.1	Installation	203
40.42	fobi.contrib.plugins.form_elements.security.recaptcha	203
40.42.1	Installation	203
40.42.1.1	Install <i>django-recaptcha</i>	203
40.42.1.2	Install <i>fobi</i> ReCAPTCHA plugin	203
40.42.2	Troubleshooting and usage limitations	204
40.42.2.1	In combination with other captcha solutions	204
40.42.3	Usage	204
40.43	fobi.contrib.plugins.form_elements.test.dummy	204
40.43.1	Installation	204
41	Form handlers	205
41.1	fobi.contrib.plugins.form_handlers.db_store	205
41.1.1	Dependencies	205
41.1.2	Installation	205
41.2	fobi.contrib.plugins.form_handlers.http_repost	206
41.2.1	Installation	206
41.3	fobi.contrib.plugins.form_handlers.mail	206
41.3.1	Installation	206
41.4	fobi.contrib.plugins.form_handlers.mail_sender	207
41.4.1	Installation	207
41.4.2	Usage	207
42	Form importers	209
42.1	mailchimp_importer	209
42.1.1	Prerequisites	209
42.1.2	Installation	209
42.1.2.1	your_project/settings.py	209
42.1.3	How it works	210
42.1.4	Development status	210
43	Themes	211
43.1	fobi.contrib.themes.bootstrap3	211
43.1.1	Installation	211
43.2	fobi.contrib.themes.djangocms_admin_style_theme	211

43.2.1	Installation	211
43.2.1.1	Install <i>djangoCMS-admin-style</i>	211
43.2.1.2	Install <i>fobi.contrib.themes.djangoCMS_admin_style_theme</i> theme	212
43.3	<i>fobi.contrib.themes.foundation5</i>	212
43.3.1	Installation	212
43.4	<i>fobi.contrib.themes.simple</i>	212
43.4.1	Installation	213
44	fobi.integration	215
45	Concepts	217
46	Templates	219
46.1	Reusable	219
46.1.1	Markdown widget	219
47	fobi.wizard	221
48	Documentation	223
48.1	Quick start	223
48.1.1	Part 1: standard Django installation	223
48.1.1.1	Installation and configuration	223
48.1.1.1.1	Install the package in your environment.	223
48.1.1.1.2	INSTALLED_APPS	223
48.1.1.1.3	TEMPLATE_CONTEXT_PROCESSORS	226
48.1.1.1.4	urlpatterns	226
48.1.1.1.5	Update the database	227
48.1.1.1.6	Specify the active theme	227
48.1.1.1.7	Permissions	227
48.1.2	Part 2: Integration with DjangoCMS	227
48.2	Release history and notes	228
48.2.1	0.19.4	228
48.2.2	0.19.3	228
48.2.3	0.19.2	228
48.2.4	0.19.1	228
48.2.5	0.19	228
48.2.6	0.18	229
48.2.7	0.17.1	230
48.2.8	0.17	230
48.2.9	0.16.4	230
48.2.10	0.16.3	230
48.2.11	0.16.2	230
48.2.12	0.16.1	230
48.2.13	0.16	231
48.2.14	0.15	231
48.2.15	0.14	231
48.2.16	0.13.10	232
48.2.17	0.13.9	232
48.2.18	0.13.8	232
48.2.19	0.13.7	232
48.2.20	0.13.6	232
48.2.21	0.13.5	232
48.2.22	0.13.4	233
48.2.23	0.13.3	233
48.2.24	0.13.2	233

48.2.25	0.13.1	233
48.2.26	0.13	233
48.2.27	0.12.20	234
48.2.28	0.12.19	234
48.2.29	0.12.18	234
48.2.30	0.12.17	234
48.2.31	0.12.16	234
48.2.32	0.12.15	235
48.2.33	0.12.14	235
48.2.34	0.12.13	235
48.2.35	0.12.12	236
48.2.36	0.12.11	236
48.2.37	0.12.10	236
48.2.38	0.12.9	236
48.2.39	0.12.8	236
48.2.40	0.12.7	237
48.2.41	0.12.6	237
48.2.42	0.12.5	237
48.2.43	0.12.4	237
48.2.44	0.12.3	237
48.2.45	0.12.2	238
48.2.46	0.12.1	238
48.2.47	0.12	238
48.2.48	0.11.13	238
48.2.49	0.11.12	238
48.2.50	0.11.11	239
48.2.51	0.11.10	239
48.2.52	0.11.9	239
48.2.53	0.11.8	239
48.2.54	0.11.7	239
48.2.55	0.11.6	239
48.2.56	0.11.5	240
48.2.57	0.11.4	240
48.2.58	0.11.3	240
48.2.59	0.11.2	240
48.2.60	0.11.1	240
48.2.61	0.11	240
48.2.62	0.10.7	241
48.2.63	0.10.6	241
48.2.64	0.10.5	241
48.2.65	0.10.4	241
48.2.66	0.10.3	241
48.2.67	0.10.2	242
48.2.68	0.10.1	242
48.2.69	0.10	242
48.2.70	0.9.17	242
48.2.71	0.9.16	243
48.2.72	0.9.15	243
48.2.73	0.9.14	243
48.2.74	0.9.13	243
48.2.75	0.9.12	243
48.2.76	0.9.11	243
48.2.77	0.9.10	244
48.2.78	0.9.9	244

48.2.79	0.9.8	244
48.2.80	0.9.7	244
48.2.81	0.9.6	244
48.2.82	0.9.5	244
48.2.83	0.9.4	245
48.2.84	0.9.3	245
48.2.85	0.9.2	245
48.2.86	0.9.1	245
48.2.87	0.9	245
48.2.88	0.8.10	246
48.2.89	0.8.9	246
48.2.90	0.8.8	246
48.2.91	0.8.7	246
48.2.92	0.8.6	246
48.2.93	0.8.5	246
48.2.94	0.8.4	247
48.2.95	0.8.3	247
48.2.96	0.8.2	247
48.2.97	0.8.1	247
48.2.98	0.8	247
48.2.98.1	New files	248
48.2.98.2	Existing files	249
48.2.98.3	Additional listings	249
48.2.98.3.1	Listing 0.8.a “tab-form-elemenets” -> “tab-form-elements”	249
48.2.98.3.2	Listing 0.8.b <i>form_view_form_entry_option_class</i> property	249
48.2.99	0.7.1	249
48.2.1000	7	250
48.2.1010	6.10	250
48.2.1020	6.9	250
48.2.1030	6.8	250
48.2.1040	6.7	250
48.2.1050	6.6	250
48.2.1060	6.5	251
48.2.1070	6.4	251
48.2.1080	6.3	251
48.2.1090	6.2	251
48.2.1100	6.1	251
48.2.1110	6	251
48.2.1120	5.19	251
48.2.1130	5.18	252
48.2.1140	5.17	252
48.2.1150	5.16	252
48.2.1160	5.15	252
48.2.1170	5.14	252
48.2.1180	5.13	252
48.2.1190	5.12	252
48.2.119.1	New files	253
48.2.119.2	Existing files	253
48.2.1200	5.11	253
48.2.1210	5.10	253
48.2.1220	5.9	253
48.2.1230	5.8	254
48.2.1240	5.7	254
48.2.1250	5.6	254

48.2.1260.5.5	254
48.2.1270.5.4	254
48.2.1280.5.3	254
48.2.1290.5.2	255
48.2.1300.5.1	255
48.2.1310.5	255
48.2.1320.4.36	255
48.2.1330.4.35	255
48.2.1340.4.34	255
48.2.1350.4.33	256
48.2.1360.4.32	256
48.2.1370.4.31	256
48.2.1380.4.30	256
48.2.1390.4.29	256
48.2.1400.4.28	257
48.2.1410.4.27	257
48.2.1420.4.26	258
48.2.1430.4.25	258
48.2.1440.4.24	258
48.2.1450.4.23	258
48.2.1460.4.22	258
48.2.1470.4.21	259
48.2.1480.4.20	259
48.2.1490.4.19	259
48.2.1500.4.18	259
48.2.1510.4.17	259
48.2.1520.4.16	260
48.2.1530.4.15	260
48.2.1540.4.14	260
48.2.1550.4.13	260
48.2.1560.4.12	260
48.2.1570.4.11	261
48.2.1580.4.10	261
48.2.1590.4.9	261
48.2.1600.4.8	261
48.2.1610.4.7	261
48.2.1620.4.6	261
48.2.1630.4.5	262
48.2.1640.4.4	262
48.2.1650.4.3	262
48.2.1660.4.2	262
48.2.1670.4.1	263
48.2.1680.4	263
48.2.1690.3.4	263
48.2.1700.3.3	264
48.2.1710.3.2	264
48.2.1720.3.1	264
48.2.1730.3	264
48.2.1740.2.1	265
48.2.1750.2	265
48.2.1760.1.6	265
48.2.1770.1.5	266
48.2.1780.1.4	266
48.2.1790.1.3	266

48.2.1800.1.2	266
48.2.1810.1.1	266
48.2.1820.1	267
48.3 Licenses	267
48.3.1 Bootstrap3	267
48.3.2 Bootstrap 3 DateTime picker	267
48.3.3 bootstrap-slider	267
48.3.4 django-autoslug	267
48.3.5 django-formtools	267
48.3.6 django-nine	268
48.3.7 django-nonefield	268
48.3.8 easy-thumbnails	268
48.3.9 Font-awesome	268
48.3.10 Foundation 5	268
48.3.11 Foundation 5 DateTime picker	269
48.3.12 jQuery Colorbox	269
48.3.13 jQuery	269
48.3.14 jQueryUI	269
48.3.15 jQuery-Slugify-Plugin	269
48.3.16 moment.js	269
48.3.17 ordereddict	270
48.3.18 Pillow	270
48.3.19 Requests	270
48.3.20 Six	270
48.3.21 Unidecode	270
48.3.22 vishap	270
48.4 fobi package	271
48.4.1 Subpackages	271
48.4.1.1 fobi.contrib package	271
48.4.1.1.1 Subpackages	271
48.4.1.1.1.1 fobi.contrib.apps package	271
48.4.1.1.1.2 Subpackages	271
48.4.1.1.1.3 fobi.contrib.apps.djangocms_integration package	271
48.4.1.1.1.4 Submodules	271
48.4.1.1.1.5 fobi.contrib.apps.djangocms_integration.apps module	271
48.4.1.1.1.6 fobi.contrib.apps.djangocms_integration.cms_plugins module	271
48.4.1.1.1.7 fobi.contrib.apps.djangocms_integration.cms_version module	271
48.4.1.1.1.8 fobi.contrib.apps.djangocms_integration.conf module	271
48.4.1.1.1.9 fobi.contrib.apps.djangocms_integration.defaults module	272
48.4.1.1.1.10 fobi.contrib.apps.djangocms_integration.helpers module	272
48.4.1.1.1.11 fobi.contrib.apps.djangocms_integration.models module	272
48.4.1.1.1.12 fobi.contrib.apps.djangocms_integration.settings module	272
48.4.1.1.1.13 Module contents	272
48.4.1.1.1.14 fobi.contrib.apps.drf_integration package	272
48.4.1.1.1.15 Subpackages	272
48.4.1.1.1.16 fobi.contrib.apps.drf_integration.form_elements package	272
48.4.1.1.1.17 Subpackages	272
48.4.1.1.1.18 fobi.contrib.apps.drf_integration.form_elements.content package	272
48.4.1.1.1.19 Subpackages	272
48.4.1.1.1.20 fobi.contrib.apps.drf_integration.form_elements.content.content_image package	272
48.4.1.1.1.21 Submodules	272
48.4.1.1.1.22 fobi.contrib.apps.drf_integration.form_elements.content.content_image.apps module	272

48.4.1.1.1.23	fobi.contrib.apps.drf_integration.form_elements.content.content_image.base module	273
48.4.1.1.1.24	fobi.contrib.apps.drf_integration.form_elements.content.content_image.fobi_integration_f module	273
48.4.1.1.1.25	Module contents	273
48.4.1.1.1.26	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url package	273
48.4.1.1.1.27	Submodules	273
48.4.1.1.1.28	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.apps module	273
48.4.1.1.1.29	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.base module	274
48.4.1.1.1.30	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.fobi_integratio module	274
48.4.1.1.1.31	Module contents	274
48.4.1.1.1.32	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown package	274
48.4.1.1.1.33	Submodules	274
48.4.1.1.1.34	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.apps module	274
48.4.1.1.1.35	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.base module	275
48.4.1.1.1.36	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.fobi_integratio module	275
48.4.1.1.1.37	Module contents	276
48.4.1.1.1.38	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext package	276
48.4.1.1.1.39	Submodules	276
48.4.1.1.1.40	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.apps module	276
48.4.1.1.1.41	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base module	276
48.4.1.1.1.42	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.fobi_integratio module	276
48.4.1.1.1.43	Module contents	277
48.4.1.1.1.44	fobi.contrib.apps.drf_integration.form_elements.content.content_text package	277
48.4.1.1.1.45	Submodules	277
48.4.1.1.1.46	fobi.contrib.apps.drf_integration.form_elements.content.content_text.apps module	277
48.4.1.1.1.47	fobi.contrib.apps.drf_integration.form_elements.content.content_text.base module	277
48.4.1.1.1.48	fobi.contrib.apps.drf_integration.form_elements.content.content_text.fobi_integratio_for module	277
48.4.1.1.1.49	Module contents	278
48.4.1.1.1.50	fobi.contrib.apps.drf_integration.form_elements.content.content_video package	278
48.4.1.1.1.51	Submodules	278
48.4.1.1.1.52	fobi.contrib.apps.drf_integration.form_elements.content.content_video.apps module	278
48.4.1.1.1.53	fobi.contrib.apps.drf_integration.form_elements.content.content_video.base module	278
48.4.1.1.1.54	fobi.contrib.apps.drf_integration.form_elements.content.content_video.fobi_integratio_for module	278

48.4.1.1.1.55	Module contents	279
48.4.1.1.1.56	Module contents	279
48.4.1.1.1.57	fobi.contrib.apps.drf_integration.form_elements.fields package . .	279
48.4.1.1.1.58	Subpackages	279
48.4.1.1.1.59	fobi.contrib.apps.drf_integration.form_elements.fields.boolean package	279
48.4.1.1.1.60	Submodules	279
48.4.1.1.1.61	fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps module	279
48.4.1.1.1.62	fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base module	279
48.4.1.1.1.63	fobi.contrib.apps.drf_integration.form_elements.fields.boolean.fobi_integration_form_ele module	280
48.4.1.1.1.64	Module contents	280
48.4.1.1.1.65	fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple package	280
48.4.1.1.1.66	Submodules	280
48.4.1.1.1.67	fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.apps module	280
48.4.1.1.1.68	fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.base module	280
48.4.1.1.1.69	fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.fobi_inte module	281
48.4.1.1.1.70	Module contents	282
48.4.1.1.1.71	fobi.contrib.apps.drf_integration.form_elements.fields.date package	282
48.4.1.1.1.72	Submodules	282
48.4.1.1.1.73	fobi.contrib.apps.drf_integration.form_elements.fields.date.apps module	282
48.4.1.1.1.74	fobi.contrib.apps.drf_integration.form_elements.fields.date.base module	282
48.4.1.1.1.75	fobi.contrib.apps.drf_integration.form_elements.fields.date.fobi_integration_form_element module	282
48.4.1.1.1.76	Module contents	283
48.4.1.1.1.77	fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down package	283
48.4.1.1.1.78	Submodules	283
48.4.1.1.1.79	fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps module	283
48.4.1.1.1.80	fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.base module	283
48.4.1.1.1.81	fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.fobi_integration_f module	283
48.4.1.1.1.82	Module contents	284
48.4.1.1.1.83	fobi.contrib.apps.drf_integration.form_elements.fields.datetime package	284
48.4.1.1.1.84	Submodules	284
48.4.1.1.1.85	fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps module	284
48.4.1.1.1.86	fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base module	284
48.4.1.1.1.87	fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_integration_form_ele module	284
48.4.1.1.1.88	Module contents	285

48.4.1.1.1.89	fobi.contrib.apps.drf_integration.form_elements.fields.decimal package	285
48.4.1.1.1.90	Submodules	285
48.4.1.1.1.91	fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps module	285
48.4.1.1.1.92	fobi.contrib.apps.drf_integration.form_elements.fields.decimal.base module	285
48.4.1.1.1.93	fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi_integration_form_elemen module	285
48.4.1.1.1.94	Module contents	286
48.4.1.1.1.95	fobi.contrib.apps.drf_integration.form_elements.fields.duration package	286
48.4.1.1.1.96	Submodules	286
48.4.1.1.1.97	fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps module	286
48.4.1.1.1.98	fobi.contrib.apps.drf_integration.form_elements.fields.duration.base module	286
48.4.1.1.1.99	fobi.contrib.apps.drf_integration.form_elements.fields.duration.fobi_integration_form_elemen module	287
48.4.1.1.1.100	Module contents	287
48.4.1.1.1.101	fobi.contrib.apps.drf_integration.form_elements.fields.email package	287
48.4.1.1.1.102	Submodules	287
48.4.1.1.1.103	fobi.contrib.apps.drf_integration.form_elements.fields.email.apps module	287
48.4.1.1.1.104	fobi.contrib.apps.drf_integration.form_elements.fields.email.base module	288
48.4.1.1.1.105	fobi.contrib.apps.drf_integration.form_elements.fields.email.fobi_integration_form_elemen module	288
48.4.1.1.1.106	Module contents	288
48.4.1.1.1.107	fobi.contrib.apps.drf_integration.form_elements.fields.file package	288
48.4.1.1.1.108	Submodules	288
48.4.1.1.1.109	fobi.contrib.apps.drf_integration.form_elements.fields.file.apps module	288
48.4.1.1.1.110	fobi.contrib.apps.drf_integration.form_elements.fields.file.base module	289
48.4.1.1.1.111	fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_integration_form_elemen module	289
48.4.1.1.1.112	Module contents	290
48.4.1.1.1.113	fobi.contrib.apps.drf_integration.form_elements.fields.float package	290
48.4.1.1.1.114	Submodules	290
48.4.1.1.1.115	fobi.contrib.apps.drf_integration.form_elements.fields.float.apps module	290
48.4.1.1.1.116	fobi.contrib.apps.drf_integration.form_elements.fields.float.base module	290
48.4.1.1.1.117	fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_integration_form_elemen module	291
48.4.1.1.1.118	Module contents	291
48.4.1.1.1.119	fobi.contrib.apps.drf_integration.form_elements.fields.hidden package	291
48.4.1.1.1.120	Submodules	291
48.4.1.1.1.121	fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps module	291

48.4.1.1.1.122	fobi.contrib.apps.drf_integration.form_elements.fields.hidden.base module	291
48.4.1.1.1.123	fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration_form_elem module	292
48.4.1.1.1.124	Module contents	292
48.4.1.1.1.125	fobi.contrib.apps.drf_integration.form_elements.fields.input package	292
48.4.1.1.1.126	Submodules	292
48.4.1.1.1.127	fobi.contrib.apps.drf_integration.form_elements.fields.input.apps module	292
48.4.1.1.1.128	fobi.contrib.apps.drf_integration.form_elements.fields.input.base module	292
48.4.1.1.1.129	fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration_form_elemen module	293
48.4.1.1.1.130	Module contents	293
48.4.1.1.1.131	fobi.contrib.apps.drf_integration.form_elements.fields.integer package	293
48.4.1.1.1.132	Submodules	293
48.4.1.1.1.133	fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps module	293
48.4.1.1.1.134	fobi.contrib.apps.drf_integration.form_elements.fields.integer.base module	293
48.4.1.1.1.135	fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_integration_form_elem module	294
48.4.1.1.1.136	Module contents	294
48.4.1.1.1.137	fobi.contrib.apps.drf_integration.form_elements.fields.ip_address package	294
48.4.1.1.1.138	Submodules	294
48.4.1.1.1.139	fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps module	294
48.4.1.1.1.140	fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base module	294
48.4.1.1.1.141	fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration_form_e module	295
48.4.1.1.1.142	Module contents	295
48.4.1.1.1.143	fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean package	295
48.4.1.1.1.144	Submodules	295
48.4.1.1.1.145	fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps module	295
48.4.1.1.1.146	fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.base module	295
48.4.1.1.1.147	fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form module	296
48.4.1.1.1.148	Module contents	296
48.4.1.1.1.149	fobi.contrib.apps.drf_integration.form_elements.fields.password package	296
48.4.1.1.1.150	Submodules	296
48.4.1.1.1.151	fobi.contrib.apps.drf_integration.form_elements.fields.password.apps module	296
48.4.1.1.1.152	fobi.contrib.apps.drf_integration.form_elements.fields.password.base module	296
48.4.1.1.1.153	fobi.contrib.apps.drf_integration.form_elements.fields.password.fobi_integration_form_e module	297
48.4.1.1.1.154	Module contents	297

48.4.1.1.1.155	fobi.contrib.apps.drf_integration.form_elements.fields.radio package	297
48.4.1.1.1.156	Submodules	297
48.4.1.1.1.157	fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps module	297
48.4.1.1.1.158	fobi.contrib.apps.drf_integration.form_elements.fields.radio.base module	297
48.4.1.1.1.159	fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_integration_form_element module	298
48.4.1.1.1.160	Module contents	298
48.4.1.1.1.161	fobi.contrib.apps.drf_integration.form_elements.fields.range_select package	298
48.4.1.1.1.162	Submodules	298
48.4.1.1.1.163	fobi.contrib.apps.drf_integration.form_elements.fields.range_select.apps module	298
48.4.1.1.1.164	fobi.contrib.apps.drf_integration.form_elements.fields.range_select.base module	298
48.4.1.1.1.165	fobi.contrib.apps.drf_integration.form_elements.fields.range_select.fobi_integration_form_element module	299
48.4.1.1.1.166	Module contents	299
48.4.1.1.1.167	fobi.contrib.apps.drf_integration.form_elements.fields.regex package	299
48.4.1.1.1.168	Submodules	299
48.4.1.1.1.169	fobi.contrib.apps.drf_integration.form_elements.fields.regex.apps module	299
48.4.1.1.1.170	fobi.contrib.apps.drf_integration.form_elements.fields.regex.base module	299
48.4.1.1.1.171	fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_integration_form_element module	300
48.4.1.1.1.172	Module contents	300
48.4.1.1.1.173	fobi.contrib.apps.drf_integration.form_elements.fields.select package	300
48.4.1.1.1.174	Submodules	300
48.4.1.1.1.175	fobi.contrib.apps.drf_integration.form_elements.fields.select.apps module	300
48.4.1.1.1.176	fobi.contrib.apps.drf_integration.form_elements.fields.select.base module	300
48.4.1.1.1.177	fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi_integration_form_element module	301
48.4.1.1.1.178	Module contents	301
48.4.1.1.1.179	fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object package	301
48.4.1.1.1.180	Submodules	301
48.4.1.1.1.181	fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.apps module	301
48.4.1.1.1.182	fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.base module	301
48.4.1.1.1.183	fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.fobi_integration_form_element module	302
48.4.1.1.1.184	Module contents	303
48.4.1.1.1.185	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple package	303
48.4.1.1.1.186	Submodules	303
48.4.1.1.1.187	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.apps module	303

48.4.1.1.1.188	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base module	303
48.4.1.1.1.189	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.fobi_integration_for module	304
48.4.1.1.1.190	Module contents	304
48.4.1.1.1.191	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects package	304
48.4.1.1.1.192	Submodules	304
48.4.1.1.1.193	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.apps module	304
48.4.1.1.1.194	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.base module	305
48.4.1.1.1.195	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.fobi module	305
48.4.1.1.1.196	Module contents	306
48.4.1.1.1.197	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max package	306
48.4.1.1.1.198	Submodules	306
48.4.1.1.1.199	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.apps module	306
48.4.1.1.1.200	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.base module	306
48.4.1.1.1.201	fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.fobi_inte module	307
48.4.1.1.1.202	Module contents	308
48.4.1.1.1.203	fobi.contrib.apps.drf_integration.form_elements.fields.slider package	308
48.4.1.1.1.204	Submodules	308
48.4.1.1.1.205	fobi.contrib.apps.drf_integration.form_elements.fields.slider.apps module	308
48.4.1.1.1.206	fobi.contrib.apps.drf_integration.form_elements.fields.slider.base module	308
48.4.1.1.1.207	fobi.contrib.apps.drf_integration.form_elements.fields.slider.fobi_integration_form_elemen module	308
48.4.1.1.1.208	Module contents	309
48.4.1.1.1.209	fobi.contrib.apps.drf_integration.form_elements.fields.slug package	309
48.4.1.1.1.210	Submodules	309
48.4.1.1.1.211	fobi.contrib.apps.drf_integration.form_elements.fields.slug.apps module	309
48.4.1.1.1.212	fobi.contrib.apps.drf_integration.form_elements.fields.slug.base module	309
48.4.1.1.1.213	fobi.contrib.apps.drf_integration.form_elements.fields.slug.fobi_integration_form_element module	309
48.4.1.1.1.214	Module contents	310
48.4.1.1.1.215	fobi.contrib.apps.drf_integration.form_elements.fields.text package	310
48.4.1.1.1.216	Submodules	310
48.4.1.1.1.217	fobi.contrib.apps.drf_integration.form_elements.fields.text.apps module	310
48.4.1.1.1.218	fobi.contrib.apps.drf_integration.form_elements.fields.text.base module	310
48.4.1.1.1.219	fobi.contrib.apps.drf_integration.form_elements.fields.text.fobi_integration_form_element module	310
48.4.1.1.1.220	Module contents	311

48.4.1.1.1.221	fobi.contrib.apps.drf_integration.form_elements.fields.textarea	
	package	311
48.4.1.1.1.222	Submodules	311
48.4.1.1.1.223	fobi.contrib.apps.drf_integration.form_elements.fields.textarea.apps	
	module	311
48.4.1.1.1.224	fobi.contrib.apps.drf_integration.form_elements.fields.textarea.base	
	module	311
48.4.1.1.1.225	fobi.contrib.apps.drf_integration.form_elements.fields.textarea.fobi_integration_form_elemen	
	module	311
48.4.1.1.1.226	Module contents	312
48.4.1.1.1.227	fobi.contrib.apps.drf_integration.form_elements.fields.time package	312
48.4.1.1.1.228	Submodules	312
48.4.1.1.1.229	fobi.contrib.apps.drf_integration.form_elements.fields.time.apps	
	module	312
48.4.1.1.1.230	fobi.contrib.apps.drf_integration.form_elements.fields.time.base	
	module	312
48.4.1.1.1.231	fobi.contrib.apps.drf_integration.form_elements.fields.time.fobi_integration_form_elemen	
	module	312
48.4.1.1.1.232	Module contents	313
48.4.1.1.1.233	fobi.contrib.apps.drf_integration.form_elements.fields.url package	313
48.4.1.1.1.234	Submodules	313
48.4.1.1.1.235	fobi.contrib.apps.drf_integration.form_elements.fields.url.apps	
	module	313
48.4.1.1.1.236	fobi.contrib.apps.drf_integration.form_elements.fields.url.base	
	module	313
48.4.1.1.1.237	fobi.contrib.apps.drf_integration.form_elements.fields.url.fobi_integration_form_elements	
	module	313
48.4.1.1.1.238	Module contents	314
48.4.1.1.1.239	Module contents	314
48.4.1.1.1.240	Module contents	314
48.4.1.1.1.241	fobi.contrib.apps.drf_integration.form_handlers package	314
48.4.1.1.1.242	Subpackages	314
48.4.1.1.1.243	fobi.contrib.apps.drf_integration.form_handlers.db_store package	314
48.4.1.1.1.244	Submodules	314
48.4.1.1.1.245	fobi.contrib.apps.drf_integration.form_handlers.db_store.apps	
	module	314
48.4.1.1.1.246	fobi.contrib.apps.drf_integration.form_handlers.db_store.base	
	module	314
48.4.1.1.1.247	fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration_form_handlers	
	module	314
48.4.1.1.1.248	Module contents	315
48.4.1.1.1.249	fobi.contrib.apps.drf_integration.form_handlers.http_repost package	315
48.4.1.1.1.250	Submodules	315
48.4.1.1.1.251	fobi.contrib.apps.drf_integration.form_handlers.http_repost.apps	
	module	315
48.4.1.1.1.252	fobi.contrib.apps.drf_integration.form_handlers.http_repost.base	
	module	315
48.4.1.1.1.253	fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handler	
	module	315
48.4.1.1.1.254	Module contents	316
48.4.1.1.1.255	fobi.contrib.apps.drf_integration.form_handlers.mail package	316
48.4.1.1.1.256	Submodules	316
48.4.1.1.1.257	fobi.contrib.apps.drf_integration.form_handlers.mail.apps module	316
48.4.1.1.1.258	fobi.contrib.apps.drf_integration.form_handlers.mail.base module	316

48.4.1.1.1.259 fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_integration_form_handlers module	316
48.4.1.1.1.260 Module contents	317
48.4.1.1.1.261 Module contents	317
48.4.1.1.1.262 Submodules	317
48.4.1.1.1.263 fobi.contrib.apps.drf_integration.apps module	317
48.4.1.1.1.264 fobi.contrib.apps.drf_integration.base module	317
48.4.1.1.1.265 fobi.contrib.apps.drf_integration.dynamic module	319
48.4.1.1.1.266 fobi.contrib.apps.drf_integration.fields module	319
48.4.1.1.1.267 fobi.contrib.apps.drf_integration.helpers module	321
48.4.1.1.1.268 fobi.contrib.apps.drf_integration.metadata module	321
48.4.1.1.1.269 fobi.contrib.apps.drf_integration.serializers module	321
48.4.1.1.1.270 fobi.contrib.apps.drf_integration.urls module	322
48.4.1.1.1.271 fobi.contrib.apps.drf_integration.utils module	322
48.4.1.1.1.272 fobi.contrib.apps.drf_integration.views module	322
48.4.1.1.1.273 Module contents	323
48.4.1.1.1.274 fobi.contrib.apps.feincms_integration package	323
48.4.1.1.1.275 Submodules	323
48.4.1.1.1.276 fobi.contrib.apps.feincms_integration.apps module	323
48.4.1.1.1.277 fobi.contrib.apps.feincms_integration.conf module	323
48.4.1.1.1.278 fobi.contrib.apps.feincms_integration.defaults module	323
48.4.1.1.1.279 fobi.contrib.apps.feincms_integration.helpers module	323
48.4.1.1.1.280 fobi.contrib.apps.feincms_integration.settings module	324
48.4.1.1.1.281 fobi.contrib.apps.feincms_integration.widgets module	324
48.4.1.1.1.282 Module contents	325
48.4.1.1.1.283 fobi.contrib.apps.mezzanine_integration package	325
48.4.1.1.1.284 Submodules	325
48.4.1.1.1.285 fobi.contrib.apps.mezzanine_integration.admin module	325
48.4.1.1.1.286 fobi.contrib.apps.mezzanine_integration.apps module	325
48.4.1.1.1.287 fobi.contrib.apps.mezzanine_integration.conf module	326
48.4.1.1.1.288 fobi.contrib.apps.mezzanine_integration.defaults module	326
48.4.1.1.1.289 fobi.contrib.apps.mezzanine_integration.helpers module	326
48.4.1.1.1.290 fobi.contrib.apps.mezzanine_integration.models module	326
48.4.1.1.1.291 fobi.contrib.apps.mezzanine_integration.page_processors module	326
48.4.1.1.1.292 fobi.contrib.apps.mezzanine_integration.settings module	326
48.4.1.1.1.293 Module contents	327
48.4.1.1.1.294 Module contents	327
48.4.1.1.1.295 fobi.contrib.plugins package	327
48.4.1.1.1.296 Subpackages	327
48.4.1.1.1.297 fobi.contrib.plugins.form_elements package	327
48.4.1.1.1.298 Subpackages	327
48.4.1.1.1.299 fobi.contrib.plugins.form_elements.content package	327
48.4.1.1.1.300 Subpackages	327
48.4.1.1.1.301 fobi.contrib.plugins.form_elements.content.content_image package	327
48.4.1.1.1.302 Submodules	327
48.4.1.1.1.303 fobi.contrib.plugins.form_elements.content.content_image.apps module	327
48.4.1.1.1.304 fobi.contrib.plugins.form_elements.content.content_image.base module	327
48.4.1.1.1.305 fobi.contrib.plugins.form_elements.content.content_image.conf module	328
48.4.1.1.1.306 fobi.contrib.plugins.form_elements.content.content_image.defaults module	328

48.4.1.1.1.307	fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements module	328
48.4.1.1.1.308	fobi.contrib.plugins.form_elements.content.content_image.forms module	329
48.4.1.1.1.309	fobi.contrib.plugins.form_elements.content.content_image.helpers module	330
48.4.1.1.1.310	fobi.contrib.plugins.form_elements.content.content_image.settings module	330
48.4.1.1.1.311	Module contents	330
48.4.1.1.1.312	fobi.contrib.plugins.form_elements.content.content_image_url package	330
48.4.1.1.1.313	Submodules	330
48.4.1.1.1.314	fobi.contrib.plugins.form_elements.content.content_image_url.apps module	330
48.4.1.1.1.315	fobi.contrib.plugins.form_elements.content.content_image_url.base module	331
48.4.1.1.1.316	fobi.contrib.plugins.form_elements.content.content_image_url.conf module	331
48.4.1.1.1.317	fobi.contrib.plugins.form_elements.content.content_image_url.defaults module	332
48.4.1.1.1.318	fobi.contrib.plugins.form_elements.content.content_image_url.fobi_form_elements module	332
48.4.1.1.1.319	fobi.contrib.plugins.form_elements.content.content_image_url.forms module	332
48.4.1.1.1.320	fobi.contrib.plugins.form_elements.content.content_image_url.settings module	333
48.4.1.1.1.321	Module contents	333
48.4.1.1.1.322	fobi.contrib.plugins.form_elements.content.content_markdown package	333
48.4.1.1.1.323	Submodules	333
48.4.1.1.1.324	fobi.contrib.plugins.form_elements.content.content_markdown.apps module	333
48.4.1.1.1.325	fobi.contrib.plugins.form_elements.content.content_markdown.base module	334
48.4.1.1.1.326	fobi.contrib.plugins.form_elements.content.content_markdown.fobi_form_elements module	335
48.4.1.1.1.327	fobi.contrib.plugins.form_elements.content.content_markdown.forms module	336
48.4.1.1.1.328	fobi.contrib.plugins.form_elements.content.content_markdown.widgets module	336
48.4.1.1.1.329	Module contents	336
48.4.1.1.1.330	fobi.contrib.plugins.form_elements.content.content_richtext package	336
48.4.1.1.1.331	Submodules	336
48.4.1.1.1.332	fobi.contrib.plugins.form_elements.content.content_richtext.apps module	336
48.4.1.1.1.333	fobi.contrib.plugins.form_elements.content.content_richtext.base module	337
48.4.1.1.1.334	fobi.contrib.plugins.form_elements.content.content_richtext.fobi_form_elements module	338
48.4.1.1.1.335	fobi.contrib.plugins.form_elements.content.content_richtext.forms module	339
48.4.1.1.1.336	fobi.contrib.plugins.form_elements.content.content_richtext.widgets module	339
48.4.1.1.1.337	Module contents	340

48.4.1.1.1.338	fobi.contrib.plugins.form_elements.content.content_text package	340
48.4.1.1.1.339	Submodules	340
48.4.1.1.1.340	fobi.contrib.plugins.form_elements.content.content_text.apps module	340
48.4.1.1.1.341	fobi.contrib.plugins.form_elements.content.content_text.base module	340
48.4.1.1.1.342	fobi.contrib.plugins.form_elements.content.content_text.conf module	341
48.4.1.1.1.343	fobi.contrib.plugins.form_elements.content.content_text.defaults module	341
48.4.1.1.1.344	fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements module	341
48.4.1.1.1.345	fobi.contrib.plugins.form_elements.content.content_text.forms module	342
48.4.1.1.1.346	fobi.contrib.plugins.form_elements.content.content_text.settings module	342
48.4.1.1.1.347	Module contents	342
48.4.1.1.1.348	fobi.contrib.plugins.form_elements.content.content_video package	342
48.4.1.1.1.349	Submodules	342
48.4.1.1.1.350	fobi.contrib.plugins.form_elements.content.content_video.apps module	342
48.4.1.1.1.351	fobi.contrib.plugins.form_elements.content.content_video.base module	343
48.4.1.1.1.352	fobi.contrib.plugins.form_elements.content.content_video.conf module	343
48.4.1.1.1.353	fobi.contrib.plugins.form_elements.content.content_video.defaults module	344
48.4.1.1.1.354	fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements module	344
48.4.1.1.1.355	fobi.contrib.plugins.form_elements.content.content_video.forms module	344
48.4.1.1.1.356	fobi.contrib.plugins.form_elements.content.content_video.settings module	345
48.4.1.1.1.357	Module contents	345
48.4.1.1.1.358	Module contents	345
48.4.1.1.1.359	fobi.contrib.plugins.form_elements.fields package	345
48.4.1.1.1.360	Subpackages	345
48.4.1.1.1.361	fobi.contrib.plugins.form_elements.fields.boolean package	345
48.4.1.1.1.362	Submodules	345
48.4.1.1.1.363	fobi.contrib.plugins.form_elements.fields.boolean.apps module	345
48.4.1.1.1.364	fobi.contrib.plugins.form_elements.fields.boolean.base module	345
48.4.1.1.1.365	fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements module	346
48.4.1.1.1.366	fobi.contrib.plugins.form_elements.fields.boolean.forms module	346
48.4.1.1.1.367	Module contents	347
48.4.1.1.1.368	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple package	347
48.4.1.1.1.369	Submodules	347
48.4.1.1.1.370	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps module	347
48.4.1.1.1.371	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.base module	347
48.4.1.1.1.372	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf module	348

48.4.1.1.1.373	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.defaults module	348
48.4.1.1.1.374	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements module	348
48.4.1.1.1.375	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms module	349
48.4.1.1.1.376	fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.settings module	350
48.4.1.1.1.377	Module contents	350
48.4.1.1.1.378	fobi.contrib.plugins.form_elements.fields.date package	350
48.4.1.1.1.379	Submodules	350
48.4.1.1.1.380	fobi.contrib.plugins.form_elements.fields.date.apps module	350
48.4.1.1.1.381	fobi.contrib.plugins.form_elements.fields.date.base module	350
48.4.1.1.1.382	fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements module	351
48.4.1.1.1.383	fobi.contrib.plugins.form_elements.fields.date.forms module	351
48.4.1.1.1.384	fobi.contrib.plugins.form_elements.fields.date.widgets module	352
48.4.1.1.1.385	Module contents	352
48.4.1.1.1.386	fobi.contrib.plugins.form_elements.fields.date_drop_down package	352
48.4.1.1.1.387	Submodules	352
48.4.1.1.1.388	fobi.contrib.plugins.form_elements.fields.date_drop_down.apps module	352
48.4.1.1.1.389	fobi.contrib.plugins.form_elements.fields.date_drop_down.base module	352
48.4.1.1.1.390	fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements module	353
48.4.1.1.1.391	fobi.contrib.plugins.form_elements.fields.date_drop_down.forms module	354
48.4.1.1.1.392	Module contents	355
48.4.1.1.1.393	fobi.contrib.plugins.form_elements.fields.datetime package	355
48.4.1.1.1.394	Submodules	355
48.4.1.1.1.395	fobi.contrib.plugins.form_elements.fields.datetime.apps module	355
48.4.1.1.1.396	fobi.contrib.plugins.form_elements.fields.datetime.base module	355
48.4.1.1.1.397	fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements module	355
48.4.1.1.1.398	fobi.contrib.plugins.form_elements.fields.datetime.forms module	356
48.4.1.1.1.399	fobi.contrib.plugins.form_elements.fields.datetime.widgets module	357
48.4.1.1.1.400	Module contents	357
48.4.1.1.1.401	fobi.contrib.plugins.form_elements.fields.decimal package	357
48.4.1.1.1.402	Submodules	357
48.4.1.1.1.403	fobi.contrib.plugins.form_elements.fields.decimal.apps module	357
48.4.1.1.1.404	fobi.contrib.plugins.form_elements.fields.decimal.base module	357
48.4.1.1.1.405	fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements module	358
48.4.1.1.1.406	fobi.contrib.plugins.form_elements.fields.decimal.forms module	358
48.4.1.1.1.407	Module contents	359
48.4.1.1.1.408	fobi.contrib.plugins.form_elements.fields.duration package	359
48.4.1.1.1.409	Submodules	359
48.4.1.1.1.410	fobi.contrib.plugins.form_elements.fields.duration.apps module	359
48.4.1.1.1.411	fobi.contrib.plugins.form_elements.fields.duration.base module	359
48.4.1.1.1.412	fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements module	360
48.4.1.1.1.413	fobi.contrib.plugins.form_elements.fields.duration.forms module	361
48.4.1.1.1.414	Module contents	362

48.4.1.1.1.415 fobi.contrib.plugins.form_elements.fields.email package	362
48.4.1.1.1.416 Submodules	362
48.4.1.1.1.417 fobi.contrib.plugins.form_elements.fields.email.apps module . . .	362
48.4.1.1.1.418 fobi.contrib.plugins.form_elements.fields.email.base module . . .	362
48.4.1.1.1.419 fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements module	362
48.4.1.1.1.420 fobi.contrib.plugins.form_elements.fields.email.forms module . .	363
48.4.1.1.1.421 Module contents	363
48.4.1.1.1.422 fobi.contrib.plugins.form_elements.fields.file package	363
48.4.1.1.1.423 Submodules	363
48.4.1.1.1.424 fobi.contrib.plugins.form_elements.fields.file.apps module	363
48.4.1.1.1.425 fobi.contrib.plugins.form_elements.fields.file.base module	364
48.4.1.1.1.426 fobi.contrib.plugins.form_elements.fields.file.conf module	364
48.4.1.1.1.427 fobi.contrib.plugins.form_elements.fields.file.defaults module . . .	365
48.4.1.1.1.428 fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements module	365
48.4.1.1.1.429 fobi.contrib.plugins.form_elements.fields.file.forms module	366
48.4.1.1.1.430 fobi.contrib.plugins.form_elements.fields.file.settings module . . .	367
48.4.1.1.1.431 Module contents	367
48.4.1.1.1.432 fobi.contrib.plugins.form_elements.fields.float package	367
48.4.1.1.1.433 Submodules	367
48.4.1.1.1.434 fobi.contrib.plugins.form_elements.fields.float.apps module	367
48.4.1.1.1.435 fobi.contrib.plugins.form_elements.fields.float.base module	367
48.4.1.1.1.436 fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements module	367
48.4.1.1.1.437 fobi.contrib.plugins.form_elements.fields.float.forms module . . .	368
48.4.1.1.1.438 Module contents	369
48.4.1.1.1.439 fobi.contrib.plugins.form_elements.fields.hidden package	369
48.4.1.1.1.440 Submodules	369
48.4.1.1.1.441 fobi.contrib.plugins.form_elements.fields.hidden.apps module . . .	369
48.4.1.1.1.442 fobi.contrib.plugins.form_elements.fields.hidden.base module . . .	369
48.4.1.1.1.443 fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements module	369
48.4.1.1.1.444 fobi.contrib.plugins.form_elements.fields.hidden.forms module . .	370
48.4.1.1.1.445 Module contents	370
48.4.1.1.1.446 fobi.contrib.plugins.form_elements.fields.input package	370
48.4.1.1.1.447 Submodules	370
48.4.1.1.1.448 fobi.contrib.plugins.form_elements.fields.input.apps module . . .	370
48.4.1.1.1.449 fobi.contrib.plugins.form_elements.fields.input.base module . . .	371
48.4.1.1.1.450 fobi.contrib.plugins.form_elements.fields.input.constants module .	371
48.4.1.1.1.451 fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements module	371
48.4.1.1.1.452 fobi.contrib.plugins.form_elements.fields.input.forms module . . .	372
48.4.1.1.1.453 Module contents	373
48.4.1.1.1.454 fobi.contrib.plugins.form_elements.fields.integer package	373
48.4.1.1.1.455 Submodules	373
48.4.1.1.1.456 fobi.contrib.plugins.form_elements.fields.integer.apps module . .	373
48.4.1.1.1.457 fobi.contrib.plugins.form_elements.fields.integer.base module . . .	373
48.4.1.1.1.458 fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements module	374
48.4.1.1.1.459 fobi.contrib.plugins.form_elements.fields.integer.forms module . .	374
48.4.1.1.1.460 Module contents	375
48.4.1.1.1.461 fobi.contrib.plugins.form_elements.fields.ip_address package . . .	375
48.4.1.1.1.462 Submodules	375

48.4.1.1.1.463	fobi.contrib.plugins.form_elements.fields.ip_address.apps module	375
48.4.1.1.1.464	fobi.contrib.plugins.form_elements.fields.ip_address.base module	375
48.4.1.1.1.465	fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements module	375
48.4.1.1.1.466	fobi.contrib.plugins.form_elements.fields.ip_address.forms module	376
48.4.1.1.1.467	Module contents	377
48.4.1.1.1.468	fobi.contrib.plugins.form_elements.fields.null_boolean package	377
48.4.1.1.1.469	Submodules	377
48.4.1.1.1.470	fobi.contrib.plugins.form_elements.fields.null_boolean.apps module	377
48.4.1.1.1.471	fobi.contrib.plugins.form_elements.fields.null_boolean.base module	377
48.4.1.1.1.472	fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements module	377
48.4.1.1.1.473	fobi.contrib.plugins.form_elements.fields.null_boolean.forms module	378
48.4.1.1.1.474	Module contents	379
48.4.1.1.1.475	fobi.contrib.plugins.form_elements.fields.password package	379
48.4.1.1.1.476	Submodules	379
48.4.1.1.1.477	fobi.contrib.plugins.form_elements.fields.password.apps module	379
48.4.1.1.1.478	fobi.contrib.plugins.form_elements.fields.password.base module	379
48.4.1.1.1.479	fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements module	379
48.4.1.1.1.480	fobi.contrib.plugins.form_elements.fields.password.forms module	380
48.4.1.1.1.481	Module contents	381
48.4.1.1.1.482	fobi.contrib.plugins.form_elements.fields.radio package	381
48.4.1.1.1.483	Submodules	381
48.4.1.1.1.484	fobi.contrib.plugins.form_elements.fields.radio.apps module	381
48.4.1.1.1.485	fobi.contrib.plugins.form_elements.fields.radio.base module	381
48.4.1.1.1.486	fobi.contrib.plugins.form_elements.fields.radio.conf module	381
48.4.1.1.1.487	fobi.contrib.plugins.form_elements.fields.radio.defaults module	382
48.4.1.1.1.488	fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements module	382
48.4.1.1.1.489	fobi.contrib.plugins.form_elements.fields.radio.forms module	382
48.4.1.1.1.490	fobi.contrib.plugins.form_elements.fields.radio.settings module	383
48.4.1.1.1.491	Module contents	383
48.4.1.1.1.492	fobi.contrib.plugins.form_elements.fields.range_select package	383
48.4.1.1.1.493	Submodules	383
48.4.1.1.1.494	fobi.contrib.plugins.form_elements.fields.range_select.apps module	383
48.4.1.1.1.495	fobi.contrib.plugins.form_elements.fields.range_select.base module	383
48.4.1.1.1.496	fobi.contrib.plugins.form_elements.fields.range_select.conf module	384
48.4.1.1.1.497	fobi.contrib.plugins.form_elements.fields.range_select.defaults module	384
48.4.1.1.1.498	fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements module	384
48.4.1.1.1.499	fobi.contrib.plugins.form_elements.fields.range_select.forms module	385
48.4.1.1.1.500	fobi.contrib.plugins.form_elements.fields.range_select.settings module	386
48.4.1.1.1.501	Module contents	386
48.4.1.1.1.502	fobi.contrib.plugins.form_elements.fields.regex package	386
48.4.1.1.1.503	Submodules	386
48.4.1.1.1.504	fobi.contrib.plugins.form_elements.fields.regex.apps module	386
48.4.1.1.1.505	fobi.contrib.plugins.form_elements.fields.regex.base module	386
48.4.1.1.1.506	fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements module	386

48.4.1.1.1.507 fobi.contrib.plugins.form_elements.fields.regex.forms module . . .	387
48.4.1.1.1.508 Module contents	387
48.4.1.1.1.509 fobi.contrib.plugins.form_elements.fields.select package	387
48.4.1.1.1.510 Submodules	387
48.4.1.1.1.511 fobi.contrib.plugins.form_elements.fields.select.apps module . . .	387
48.4.1.1.1.512 fobi.contrib.plugins.form_elements.fields.select.base module . . .	388
48.4.1.1.1.513 fobi.contrib.plugins.form_elements.fields.select.conf module . . .	388
48.4.1.1.1.514 fobi.contrib.plugins.form_elements.fields.select.defaults module .	389
48.4.1.1.1.515 fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements module	389
48.4.1.1.1.516 fobi.contrib.plugins.form_elements.fields.select.forms module . .	389
48.4.1.1.1.517 fobi.contrib.plugins.form_elements.fields.select.settings module .	390
48.4.1.1.1.518 Module contents	390
48.4.1.1.1.519 fobi.contrib.plugins.form_elements.fields.select_model_object package	390
48.4.1.1.1.520 Submodules	390
48.4.1.1.1.521 fobi.contrib.plugins.form_elements.fields.select_model_object.apps module	390
48.4.1.1.1.522 fobi.contrib.plugins.form_elements.fields.select_model_object.base module	390
48.4.1.1.1.523 fobi.contrib.plugins.form_elements.fields.select_model_object.conf module	391
48.4.1.1.1.524 fobi.contrib.plugins.form_elements.fields.select_model_object.defaults module	391
48.4.1.1.1.525 fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements module	391
48.4.1.1.1.526 fobi.contrib.plugins.form_elements.fields.select_model_object.forms module	392
48.4.1.1.1.527 fobi.contrib.plugins.form_elements.fields.select_model_object.settings module	393
48.4.1.1.1.528 Module contents	393
48.4.1.1.1.529 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object package	393
48.4.1.1.1.530 Submodules	393
48.4.1.1.1.531 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps module	393
48.4.1.1.1.532 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base module	393
48.4.1.1.1.533 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf module	394
48.4.1.1.1.534 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults module	394
48.4.1.1.1.535 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.fobi_form_elements module	394
48.4.1.1.1.536 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms module	395
48.4.1.1.1.537 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.settings module	395
48.4.1.1.1.538 Module contents	395
48.4.1.1.1.539 fobi.contrib.plugins.form_elements.fields.select_multiple package	395
48.4.1.1.1.540 Submodules	395
48.4.1.1.1.541 fobi.contrib.plugins.form_elements.fields.select_multiple.apps module	395

48.4.1.1.1.542	fobi.contrib.plugins.form_elements.fields.select_multiple.base module	396
48.4.1.1.1.543	fobi.contrib.plugins.form_elements.fields.select_multiple.conf module	396
48.4.1.1.1.544	fobi.contrib.plugins.form_elements.fields.select_multiple.defaults module	397
48.4.1.1.1.545	fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements module	397
48.4.1.1.1.546	fobi.contrib.plugins.form_elements.fields.select_multiple.forms module	398
48.4.1.1.1.547	fobi.contrib.plugins.form_elements.fields.select_multiple.settings module	399
48.4.1.1.1.548	Module contents	399
48.4.1.1.1.549	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects package	399
48.4.1.1.1.550	Submodules	399
48.4.1.1.1.551	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps module	399
48.4.1.1.1.552	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.base module	399
48.4.1.1.1.553	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf module	400
48.4.1.1.1.554	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults module	400
48.4.1.1.1.555	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_eleme module	400
48.4.1.1.1.556	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms module	401
48.4.1.1.1.557	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.settings module	401
48.4.1.1.1.558	Module contents	401
48.4.1.1.1.559	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects package	401
48.4.1.1.1.560	Submodules	401
48.4.1.1.1.561	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps module	401
48.4.1.1.1.562	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.base module	402
48.4.1.1.1.563	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf module	402
48.4.1.1.1.564	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults module	403
48.4.1.1.1.565	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_ module	403
48.4.1.1.1.566	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms module	403
48.4.1.1.1.567	fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.settings module	404
48.4.1.1.1.568	Module contents	404
48.4.1.1.1.569	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max package	404
48.4.1.1.1.570	Submodules	404
48.4.1.1.1.571	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.apps module	404

48.4.1.1.1.572	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base module	404
48.4.1.1.1.573	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.conf module	405
48.4.1.1.1.574	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.defaults module	405
48.4.1.1.1.575	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fields module	405
48.4.1.1.1.576	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements module	406
48.4.1.1.1.577	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms module	407
48.4.1.1.1.578	fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.settings module	408
48.4.1.1.1.579	Module contents	408
48.4.1.1.1.580	fobi.contrib.plugins.form_elements.fields.slider package	408
48.4.1.1.1.581	Submodules	408
48.4.1.1.1.582	fobi.contrib.plugins.form_elements.fields.slider.apps module	408
48.4.1.1.1.583	fobi.contrib.plugins.form_elements.fields.slider.base module	408
48.4.1.1.1.584	fobi.contrib.plugins.form_elements.fields.slider.conf module	409
48.4.1.1.1.585	fobi.contrib.plugins.form_elements.fields.slider.constants module	409
48.4.1.1.1.586	fobi.contrib.plugins.form_elements.fields.slider.defaults module	409
48.4.1.1.1.587	fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements module	409
48.4.1.1.1.588	fobi.contrib.plugins.form_elements.fields.slider.forms module	410
48.4.1.1.1.589	fobi.contrib.plugins.form_elements.fields.slider.helpers module	411
48.4.1.1.1.590	fobi.contrib.plugins.form_elements.fields.slider.settings module	411
48.4.1.1.1.591	fobi.contrib.plugins.form_elements.fields.slider.widgets module	411
48.4.1.1.1.592	Module contents	411
48.4.1.1.1.593	fobi.contrib.plugins.form_elements.fields.slug package	411
48.4.1.1.1.594	Submodules	411
48.4.1.1.1.595	fobi.contrib.plugins.form_elements.fields.slug.apps module	411
48.4.1.1.1.596	fobi.contrib.plugins.form_elements.fields.slug.base module	412
48.4.1.1.1.597	fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements module	412
48.4.1.1.1.598	fobi.contrib.plugins.form_elements.fields.slug.forms module	412
48.4.1.1.1.599	Module contents	413
48.4.1.1.1.600	fobi.contrib.plugins.form_elements.fields.text package	413
48.4.1.1.1.601	Submodules	413
48.4.1.1.1.602	fobi.contrib.plugins.form_elements.fields.text.apps module	413
48.4.1.1.1.603	fobi.contrib.plugins.form_elements.fields.text.base module	413
48.4.1.1.1.604	fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements module	414
48.4.1.1.1.605	fobi.contrib.plugins.form_elements.fields.text.forms module	414
48.4.1.1.1.606	Module contents	415
48.4.1.1.1.607	fobi.contrib.plugins.form_elements.fields.textarea package	415
48.4.1.1.1.608	Submodules	415
48.4.1.1.1.609	fobi.contrib.plugins.form_elements.fields.textarea.apps module	415
48.4.1.1.1.610	fobi.contrib.plugins.form_elements.fields.textarea.base module	415
48.4.1.1.1.611	fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements module	415
48.4.1.1.1.612	fobi.contrib.plugins.form_elements.fields.textarea.forms module	416
48.4.1.1.1.613	Module contents	416
48.4.1.1.1.614	fobi.contrib.plugins.form_elements.fields.time package	416

48.4.1.1.1.615	Submodules	416
48.4.1.1.1.616	fobi.contrib.plugins.form_elements.fields.time.apps module	416
48.4.1.1.1.617	fobi.contrib.plugins.form_elements.fields.time.base module	417
48.4.1.1.1.618	fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements module	417
48.4.1.1.1.619	fobi.contrib.plugins.form_elements.fields.time.forms module	418
48.4.1.1.1.620	Module contents	418
48.4.1.1.1.621	fobi.contrib.plugins.form_elements.fields.url package	418
48.4.1.1.1.622	Submodules	418
48.4.1.1.1.623	fobi.contrib.plugins.form_elements.fields.url.apps module	418
48.4.1.1.1.624	fobi.contrib.plugins.form_elements.fields.url.base module	419
48.4.1.1.1.625	fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements module	419
48.4.1.1.1.626	fobi.contrib.plugins.form_elements.fields.url.forms module	419
48.4.1.1.1.627	Module contents	420
48.4.1.1.1.628	Module contents	420
48.4.1.1.1.629	fobi.contrib.plugins.form_elements.security package	420
48.4.1.1.1.630	Subpackages	420
48.4.1.1.1.631	fobi.contrib.plugins.form_elements.security.captcha package	420
48.4.1.1.1.632	Submodules	420
48.4.1.1.1.633	fobi.contrib.plugins.form_elements.security.captcha.apps module	420
48.4.1.1.1.634	fobi.contrib.plugins.form_elements.security.captcha.base module	421
48.4.1.1.1.635	fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements module	421
48.4.1.1.1.636	fobi.contrib.plugins.form_elements.security.captcha.forms module	421
48.4.1.1.1.637	Module contents	422
48.4.1.1.1.638	fobi.contrib.plugins.form_elements.security.honeypot package	422
48.4.1.1.1.639	Submodules	422
48.4.1.1.1.640	fobi.contrib.plugins.form_elements.security.honeypot.apps module	422
48.4.1.1.1.641	fobi.contrib.plugins.form_elements.security.honeypot.base module	422
48.4.1.1.1.642	fobi.contrib.plugins.form_elements.security.honeypot.conf module	423
48.4.1.1.1.643	fobi.contrib.plugins.form_elements.security.honeypot.defaults module	423
48.4.1.1.1.644	fobi.contrib.plugins.form_elements.security.honeypot.fields module	423
48.4.1.1.1.645	fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements module	423
48.4.1.1.1.646	fobi.contrib.plugins.form_elements.security.honeypot.forms module	424
48.4.1.1.1.647	fobi.contrib.plugins.form_elements.security.honeypot.settings module	424
48.4.1.1.1.648	Module contents	425
48.4.1.1.1.649	fobi.contrib.plugins.form_elements.security.recaptcha package	425
48.4.1.1.1.650	Submodules	425
48.4.1.1.1.651	fobi.contrib.plugins.form_elements.security.recaptcha.apps module	425
48.4.1.1.1.652	fobi.contrib.plugins.form_elements.security.recaptcha.base module	425
48.4.1.1.1.653	fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements module	425
48.4.1.1.1.654	fobi.contrib.plugins.form_elements.security.recaptcha.forms module	426
48.4.1.1.1.655	Module contents	426
48.4.1.1.1.656	Module contents	426
48.4.1.1.1.657	fobi.contrib.plugins.form_elements.test package	426
48.4.1.1.1.658	Subpackages	426
48.4.1.1.1.659	fobi.contrib.plugins.form_elements.test.dummy package	426
48.4.1.1.1.660	Submodules	426
48.4.1.1.1.661	fobi.contrib.plugins.form_elements.test.dummy.apps module	426

48.4.1.1.1.662	fobi.contrib.plugins.form_elements.test.dummy.base module . . .	427
48.4.1.1.1.663	fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements module	427
48.4.1.1.1.664	fobi.contrib.plugins.form_elements.test.dummy.widgets module .	427
48.4.1.1.1.665	Module contents	428
48.4.1.1.1.666	Module contents	428
48.4.1.1.1.667	Module contents	428
48.4.1.1.1.668	fobi.contrib.plugins.form_handlers package	428
48.4.1.1.1.669	Subpackages	428
48.4.1.1.1.670	fobi.contrib.plugins.form_handlers.db_store package	428
48.4.1.1.1.671	Subpackages	428
48.4.1.1.1.672	fobi.contrib.plugins.form_handlers.db_store.migrations package .	428
48.4.1.1.1.673	Submodules	428
48.4.1.1.1.674	fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial module	428
48.4.1.1.1.675	fobi.contrib.plugins.form_handlers.db_store.migrations.0002_savedformwizarddataentry module	428
48.4.1.1.1.676	Module contents	429
48.4.1.1.1.677	fobi.contrib.plugins.form_handlers.db_store.urls package	429
48.4.1.1.1.678	Submodules	429
48.4.1.1.1.679	fobi.contrib.plugins.form_handlers.db_store.urls.form_handlers module	429
48.4.1.1.1.680	fobi.contrib.plugins.form_handlers.db_store.urls.form_wizard_handlers module	429
48.4.1.1.1.681	Module contents	429
48.4.1.1.1.682	Submodules	429
48.4.1.1.1.683	fobi.contrib.plugins.form_handlers.db_store.admin module	429
48.4.1.1.1.684	fobi.contrib.plugins.form_handlers.db_store.apps module	430
48.4.1.1.1.685	fobi.contrib.plugins.form_handlers.db_store.base module	430
48.4.1.1.1.686	fobi.contrib.plugins.form_handlers.db_store.callbacks module . .	432
48.4.1.1.1.687	fobi.contrib.plugins.form_handlers.db_store.conf module	432
48.4.1.1.1.688	fobi.contrib.plugins.form_handlers.db_store.defaults module . . .	432
48.4.1.1.1.689	fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers module	432
48.4.1.1.1.690	fobi.contrib.plugins.form_handlers.db_store.helpers module	434
48.4.1.1.1.691	fobi.contrib.plugins.form_handlers.db_store.models module	434
48.4.1.1.1.692	fobi.contrib.plugins.form_handlers.db_store.settings module . . .	437
48.4.1.1.1.693	fobi.contrib.plugins.form_handlers.db_store.views module	437
48.4.1.1.1.694	fobi.contrib.plugins.form_handlers.db_store.widgets module . . .	438
48.4.1.1.1.695	Module contents	438
48.4.1.1.1.696	fobi.contrib.plugins.form_handlers.http_repost package	438
48.4.1.1.1.697	Submodules	438
48.4.1.1.1.698	fobi.contrib.plugins.form_handlers.http_repost.apps module	438
48.4.1.1.1.699	fobi.contrib.plugins.form_handlers.http_repost.base module	438
48.4.1.1.1.700	fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers module	440
48.4.1.1.1.701	fobi.contrib.plugins.form_handlers.http_repost.forms module . . .	441
48.4.1.1.1.702	Module contents	441
48.4.1.1.1.703	fobi.contrib.plugins.form_handlers.mail package	441
48.4.1.1.1.704	Submodules	441
48.4.1.1.1.705	fobi.contrib.plugins.form_handlers.mail.apps module	441
48.4.1.1.1.706	fobi.contrib.plugins.form_handlers.mail.base module	442
48.4.1.1.1.707	fobi.contrib.plugins.form_handlers.mail.callbacks module	443
48.4.1.1.1.708	fobi.contrib.plugins.form_handlers.mail.conf module	443

48.4.1.1.1.709 fobi.contrib.plugins.form_handlers.mail.defaults module	444
48.4.1.1.1.710 fobi.contrib.plugins.form_handlers.mail.fields module	444
48.4.1.1.1.711 fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers module	444
48.4.1.1.1.712 fobi.contrib.plugins.form_handlers.mail.forms module	445
48.4.1.1.1.713 fobi.contrib.plugins.form_handlers.mail.helpers module	446
48.4.1.1.1.714 fobi.contrib.plugins.form_handlers.mail.mixins module	446
48.4.1.1.1.715 fobi.contrib.plugins.form_handlers.mail.settings module	447
48.4.1.1.1.716 fobi.contrib.plugins.form_handlers.mail.widgets module	447
48.4.1.1.1.717 Module contents	447
48.4.1.1.1.718 Module contents	447
48.4.1.1.1.719 fobi.contrib.plugins.form_importers package	447
48.4.1.1.1.720 Subpackages	447
48.4.1.1.1.721 fobi.contrib.plugins.form_importers.mailchimp_importer package	447
48.4.1.1.1.722 Submodules	447
48.4.1.1.1.723 fobi.contrib.plugins.form_importers.mailchimp_importer.apps module	447
48.4.1.1.1.724 fobi.contrib.plugins.form_importers.mailchimp_importer.base module	448
48.4.1.1.1.725 fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers module	448
48.4.1.1.1.726 fobi.contrib.plugins.form_importers.mailchimp_importer.forms module	449
48.4.1.1.1.727 fobi.contrib.plugins.form_importers.mailchimp_importer.views module	450
48.4.1.1.1.728 Module contents	451
48.4.1.1.1.729 Module contents	451
48.4.1.1.1.730 Module contents	451
48.4.1.1.1.731 fobi.contrib.themes package	451
48.4.1.1.1.732 Subpackages	451
48.4.1.1.1.733 fobi.contrib.themes.bootstrap3 package	451
48.4.1.1.1.734 Subpackages	451
48.4.1.1.1.735 fobi.contrib.themes.bootstrap3.widgets package	451
48.4.1.1.1.736 Subpackages	451
48.4.1.1.1.737 fobi.contrib.themes.bootstrap3.widgets.form_elements package	451
48.4.1.1.1.738 Subpackages	451
48.4.1.1.1.739 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widgets package	451
48.4.1.1.1.740 Submodules	451
48.4.1.1.1.741 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widgets module	451
48.4.1.1.1.742 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widgets module	452
48.4.1.1.1.743 Module contents	452
48.4.1.1.1.744 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widgets package	452
48.4.1.1.1.745 Submodules	452
48.4.1.1.1.746 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widgets module	452
48.4.1.1.1.747 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widgets module	452
48.4.1.1.1.748 Module contents	453
48.4.1.1.1.749 fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widgets package	453
48.4.1.1.1.750 Submodules	453

48.4.1.1.1.751	fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps module	453
48.4.1.1.1.752	fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_ module	453
48.4.1.1.1.753	Module contents	453
48.4.1.1.1.754	fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget package	453
48.4.1.1.1.755	Submodules	453
48.4.1.1.1.756	fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps module	453
48.4.1.1.1.757	fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_f module	454
48.4.1.1.1.758	Module contents	454
48.4.1.1.1.759	fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget package	454
48.4.1.1.1.760	Submodules	454
48.4.1.1.1.761	fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps module	454
48.4.1.1.1.762	fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_for module	455
48.4.1.1.1.763	Module contents	455
48.4.1.1.1.764	fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget package	455
48.4.1.1.1.765	Submodules	455
48.4.1.1.1.766	fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps module	455
48.4.1.1.1.767	fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_for module	455
48.4.1.1.1.768	Module contents	456
48.4.1.1.1.769	Module contents	456
48.4.1.1.1.770	Module contents	456
48.4.1.1.1.771	Submodules	456
48.4.1.1.1.772	fobi.contrib.themes.bootstrap3.apps module	456
48.4.1.1.1.773	fobi.contrib.themes.bootstrap3.fobi_themes module	456
48.4.1.1.1.774	Module contents	458
48.4.1.1.1.775	fobi.contrib.themes.djangocms_admin_style_theme package	458
48.4.1.1.1.776	Subpackages	458
48.4.1.1.1.777	fobi.contrib.themes.djangocms_admin_style_theme.widgets package	458
48.4.1.1.1.778	Subpackages	458
48.4.1.1.1.779	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements package	458
48.4.1.1.1.780	Subpackages	458
48.4.1.1.1.781	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richte package	458
48.4.1.1.1.782	Submodules	458
48.4.1.1.1.783	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richte module	458
48.4.1.1.1.784	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richte module	459
48.4.1.1.1.785	Module contents	459
48.4.1.1.1.786	Module contents	459
48.4.1.1.1.787	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers package	459
48.4.1.1.1.788	Subpackages	459

48.4.1.1.1.789	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store	
	package	459
48.4.1.1.1.790	Submodules	459
48.4.1.1.1.791	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps	
	module	459
48.4.1.1.1.792	fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_	
	module	459
48.4.1.1.1.793	Module contents	460
48.4.1.1.1.794	Module contents	460
48.4.1.1.1.795	Module contents	460
48.4.1.1.1.796	Submodules	460
48.4.1.1.1.797	fobi.contrib.themes.djangocms_admin_style_theme.apps module	460
48.4.1.1.1.798	fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes	
	module	460
48.4.1.1.1.799	Module contents	462
48.4.1.1.1.800	fobi.contrib.themes.foundation5 package	462
48.4.1.1.1.801	Subpackages	462
48.4.1.1.1.802	fobi.contrib.themes.foundation5.widgets package	462
48.4.1.1.1.803	Subpackages	462
48.4.1.1.1.804	fobi.contrib.themes.foundation5.widgets.form_elements package	462
48.4.1.1.1.805	Subpackages	462
48.4.1.1.1.806	fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_wi	
	package	462
48.4.1.1.1.807	Submodules	462
48.4.1.1.1.808	fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_wi	
	module	462
48.4.1.1.1.809	fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_wi	
	module	463
48.4.1.1.1.810	Module contents	463
48.4.1.1.1.811	fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget	
	package	463
48.4.1.1.1.812	Submodules	463
48.4.1.1.1.813	fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps	
	module	463
48.4.1.1.1.814	fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_for	
	module	463
48.4.1.1.1.815	Module contents	464
48.4.1.1.1.816	fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget	
	package	464
48.4.1.1.1.817	Submodules	464
48.4.1.1.1.818	fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.app	
	module	464
48.4.1.1.1.819	fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.fobi	
	module	464
48.4.1.1.1.820	Module contents	464
48.4.1.1.1.821	fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget	
	package	464
48.4.1.1.1.822	Submodules	464
48.4.1.1.1.823	fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps	
	module	464
48.4.1.1.1.824	fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_	
	module	465
48.4.1.1.1.825	Module contents	465
48.4.1.1.1.826	Module contents	465

48.4.1.1.1.827	fobi.contrib.themes.foundation5.widgets.form_handlers package	465
48.4.1.1.1.828	Subpackages	465
48.4.1.1.1.829	fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget package	465
48.4.1.1.1.830	Submodules	465
48.4.1.1.1.831	fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps module	465
48.4.1.1.1.832	fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi module	466
48.4.1.1.1.833	Module contents	466
48.4.1.1.1.834	Module contents	466
48.4.1.1.1.835	Module contents	466
48.4.1.1.1.836	Submodules	466
48.4.1.1.1.837	fobi.contrib.themes.foundation5.apps module	466
48.4.1.1.1.838	fobi.contrib.themes.foundation5.fobi_themes module	466
48.4.1.1.1.839	Module contents	468
48.4.1.1.1.840	fobi.contrib.themes.simple package	468
48.4.1.1.1.841	Subpackages	468
48.4.1.1.1.842	fobi.contrib.themes.simple.widgets package	468
48.4.1.1.1.843	Subpackages	468
48.4.1.1.1.844	fobi.contrib.themes.simple.widgets.form_elements package	468
48.4.1.1.1.845	Subpackages	468
48.4.1.1.1.846	fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget package	468
48.4.1.1.1.847	Submodules	468
48.4.1.1.1.848	fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.apps module	468
48.4.1.1.1.849	fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi module	469
48.4.1.1.1.850	Module contents	469
48.4.1.1.1.851	Module contents	469
48.4.1.1.1.852	fobi.contrib.themes.simple.widgets.form_handlers package	469
48.4.1.1.1.853	Subpackages	469
48.4.1.1.1.854	fobi.contrib.themes.simple.widgets.form_handlers.db_store package	469
48.4.1.1.1.855	Submodules	469
48.4.1.1.1.856	fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps module	469
48.4.1.1.1.857	fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements module	469
48.4.1.1.1.858	Module contents	470
48.4.1.1.1.859	Module contents	470
48.4.1.1.1.860	Module contents	470
48.4.1.1.1.861	Submodules	470
48.4.1.1.1.862	fobi.contrib.themes.simple.apps module	470
48.4.1.1.1.863	fobi.contrib.themes.simple.fobi_themes module	470
48.4.1.1.1.864	Module contents	472
48.4.1.1.1.865	Module contents	472
48.4.1.1.2	Module contents	472
48.4.1.2	fobi.integration package	472
48.4.1.2.1	Submodules	472
48.4.1.2.2	fobi.integration.helpers module	472
48.4.1.2.3	fobi.integration.processors module	472
48.4.1.2.4	Module contents	473
48.4.1.3	fobi.management package	473

48.4.1.3.1	Subpackages	473
48.4.1.3.1.1	fobi.management.commands package	473
48.4.1.3.1.2	Submodules	473
48.4.1.3.1.3	fobi.management.commands.fobi_find_broken_entries module	473
48.4.1.3.1.4	fobi.management.commands.fobi_sync_plugins module	474
48.4.1.3.1.5	fobi.management.commands.fobi_update_plugin_data module	474
48.4.1.3.1.6	Module contents	474
48.4.1.3.2	Module contents	474
48.4.1.4	fobi.migrations package	474
48.4.1.4.1	Submodules	474
48.4.1.4.2	fobi.migrations.0001_initial module	474
48.4.1.4.3	fobi.migrations.0002_auto_20150912_1744 module	476
48.4.1.4.4	fobi.migrations.0003_auto_20160517_1005 module	476
48.4.1.4.5	fobi.migrations.0004_auto_20160906_1513 module	476
48.4.1.4.6	fobi.migrations.0005_auto_20160908_1457 module	477
48.4.1.4.7	fobi.migrations.0006_auto_20160911_1549 module	477
48.4.1.4.8	fobi.migrations.0007_auto_20160926_1652 module	477
48.4.1.4.9	fobi.migrations.0008_formwizardhandlerentry module	478
48.4.1.4.10	fobi.migrations.0009_formwizardentry_wizard_type module	478
48.4.1.4.11	fobi.migrations.0010_formwizardhandler module	478
48.4.1.4.12	fobi.migrations.0011_formentry_title module	478
48.4.1.4.13	fobi.migrations.0012_auto_20161109_1550 module	479
48.4.1.4.14	fobi.migrations.0013_formwizardentry_show_all_navigation_buttons module	479
48.4.1.4.15	fobi.migrations.0014_auto_20170516_1413 module	479
48.4.1.4.16	Module contents	479
48.4.1.5	fobi.reusable package	479
48.4.1.5.1	Subpackages	479
48.4.1.5.1.1	fobi.reusable.markdown_widget package	479
48.4.1.5.1.2	Submodules	479
48.4.1.5.1.3	fobi.reusable.markdown_widget.helpers module	479
48.4.1.5.1.4	fobi.reusable.markdown_widget.widgets module	480
48.4.1.5.1.5	Module contents	480
48.4.1.5.2	Module contents	480
48.4.1.6	fobi.templatetags package	480
48.4.1.6.1	Submodules	480
48.4.1.6.2	fobi.templatetags.fobi_tags module	480
48.4.1.6.3	fobi.templatetags.future_compat module	482
48.4.1.6.4	Module contents	482
48.4.1.7	fobi.tests package	482
48.4.1.7.1	Submodules	482
48.4.1.7.2	fobi.tests.base module	482
48.4.1.7.3	fobi.tests.constants module	483
48.4.1.7.4	fobi.tests.core module	483
48.4.1.7.5	fobi.tests.data module	483
48.4.1.7.6	fobi.tests.helpers module	483
48.4.1.7.7	fobi.tests.test_browser_build_dynamic_forms module	484
48.4.1.7.8	fobi.tests.test_core module	485
48.4.1.7.9	fobi.tests.test_drf_integration module	485
48.4.1.7.10	fobi.tests.test_dynamic_forms module	486
48.4.1.7.11	fobi.tests.test_feincms_integration module	486
48.4.1.7.12	fobi.tests.test_form_importers_mailchimp module	486
48.4.1.7.13	fobi.tests.test_sortable_dict module	487
48.4.1.7.14	Module contents	487

48.4.1.8	fobi.urls package	487
48.4.1.8.1	Submodules	487
48.4.1.8.2	fobi.urls.edit module	487
48.4.1.8.3	fobi.urls.view module	487
48.4.1.8.4	Module contents	487
48.4.1.9	fobi.wizard package	487
48.4.1.9.1	Subpackages	487
48.4.1.9.1.1	fobi.wizard.views package	487
48.4.1.9.1.2	Submodules	487
48.4.1.9.1.3	fobi.wizard.views.dynamic module	487
48.4.1.9.1.4	fobi.wizard.views.views module	492
48.4.1.9.1.5	Module contents	493
48.4.1.9.2	Module contents	493
48.4.2	Submodules	493
48.4.3	fobi.admin module	493
48.4.4	fobi.app module	498
48.4.5	fobi.apps module	498
48.4.6	fobi.base module	498
48.4.7	fobi.conf module	517
48.4.8	fobi.constants module	518
48.4.9	fobi.context_processors module	518
48.4.10	fobi.data_structures module	518
48.4.11	fobi.decorators module	520
48.4.12	fobi.defaults module	521
48.4.13	fobi.discover module	521
48.4.14	fobi.dynamic module	521
48.4.15	fobi.exceptions module	522
48.4.16	fobi.form_importers module	523
48.4.17	fobi.form_utils module	524
48.4.18	fobi.forms module	524
48.4.19	fobi.helpers module	530
48.4.20	fobi.models module	536
48.4.21	fobi.settings module	550
48.4.22	fobi.test module	551
48.4.23	fobi.utils module	551
48.4.24	fobi.validators module	554
48.4.25	fobi.views module	554
48.4.26	fobi.widgets module	554
48.4.27	Module contents	555
49	Indices and tables	557
	Bibliography	559
	Python Module Index	561
	Index	573

django-fobi (or just *fobi*) is a customisable, modular, user- and developer- friendly form generator/builder application for Django. With *fobi* you can build Django forms using an intuitive GUI, save or mail posted form data or even export forms into JSON format and import them on other instances. API allows you to build your own form elements and form handlers (mechanisms for handling the submitted form data).

PREREQUISITES

- Django 2.2, 3.0, 3.1, 3.2 and 4.0.
- Python 3.6, 3.7, 3.8 and 3.9.

KEY CONCEPTS

- Each form consists of elements. Form elements are divided into groups:
 - (a) form fields (input field, textarea, hidden field, file field, etc.).
 - (b) content (presentational) elements (text, image, embed video, etc.).
 - (c) security elements (captcha, etc).
- Number of form elements is not limited.
- Each form may contain handlers. Handler processes the form data (for example, saves it or mails it). Number of the handlers is not limited.
- Both form elements and form handlers are made with Django permission system in mind.
- As an addition to form handlers, form callbacks are implemented. Form callbacks are fired on various stages of pre- and post-processing the form data (on POST). Form callbacks do not make use of permission system (unless you intentionally do so in the code of your callback) and are fired for all forms (unlike form handlers, that are executed only if assigned).
- Each plugin (form element or form handler) or a callback - is a Django micro-app.
- In addition for form element and form handler plugins, integration form element and integration form handler plugins are implemented for integration with diverse third-party apps and frameworks (such as Django REST framework).

Note, that *django-fobi* does not require django-admin and administrative rights/permissions to access the UI, although almost seamless integration with django-admin is implemented through the `simple` theme.

MAIN FEATURES AND HIGHLIGHTS

- User-friendly GUI to quickly build forms.
- Large variety of *Bundled form element plugins*. Most of the Django fields are supported. *HTML5 fields* are supported as well.
- *Form wizards*. Combine your forms into wizards. Form wizards may contain handlers. Handler processes the form wizard data (for example, saves it or mails it). Number of the form wizard handlers is not limited.
- Forms can be automatically enabled/disabled based on dates (start date, end date).
- Anti-spam solutions like *CAPTCHA*, *ReCAPTCHA*, *Honeypot* or *Invisible reCAPTCHA* come out of the box (*CAPTCHA* and *ReCAPTCHA* do require additional third-party apps to be installed; *Invisible reCAPTCHA* doesn't).
- In addition to standard form elements, there are cosmetic (presentational) form elements (for adding a piece of text, image or a embed video) alongside standard form elements.
- Data handling in plugins (form handlers). Save the data, mail it to some address or re-post it to some other endpoint. See the *Bundled form handler plugins* for more information.
- Developer-friendly API, which allows to edit existing or build new form fields and handlers without touching the core.
- Support for custom user model.
- Class based views (and class-based permissions). Forms have an owner (`auth.User`). Default permissions are set for the owner, but class-based views provide a lot of freedom and can be easily customized.
- *Theming*. There are 4 ready to use *Bundled themes*: “Bootstrap 3”, “Foundation 5”, “Simple” (with editing interface in style of Django admin) and “DjangoCMS admin style” theme (which is another simple theme with editing interface in style of `django-cms-admin-style`).
- Implemented *integration with Django REST framework*.
- Implemented *integration with Wagtail* (in a form of a Wagtail page).
- Implemented *integration with FeinCMS* (in a form of a FeinCMS page widget).
- Implemented *integration with DjangoCMS* (in a form of a DjangoCMS page plugin).
- Implemented *integration with Mezzanine* (in a form of a Mezzanine page).
- Reordering of form elements using drag-n-drop.
- Data export (*DB store* form handler plugin) into XLS/CSV format.
- *Dynamic initial values* for form elements.
- Import/export forms to/from JSON format.
- Import forms from MailChimp using *mailchimp importer*.

ROADMAP

Some of the upcoming/in-development features/improvements are:

- Implement disabling forms based on dates.
- Cloning of forms.
- JSON schema support.
- Webpack integration.
- Improved Django REST framework OPTIONS.
- Bootstrap 4 support.
- Foundation 6 support.

See the [TODOS](#) for the full list of planned-, pending- in-development- or to-be-implemented features.

SOME SCREENSHOTS

See the documentation for some screen shots:

- [ReadTheDocs](#)

6.1 Live demo

See the [live demo app](#) on Heroku. Additionally, see the [Django REST framework integration demo](#).

Credentials:

- username: test_user
- password: test_user

6.2 Run demo locally

In order to be able to quickly evaluate the `django-fobi`, a demo app (with a quick installer) has been created (works on Ubuntu/Debian, may work on other Linux systems as well, although not guaranteed). Follow the instructions below for having the demo running within a minute.

Grab the latest `django_fobi_example_app_installer.sh`:

```
wget https://raw.githubusercontent.com/barseghyanartur/django-fobi/stable/examples/django_fobi_
↳example_app_installer.sh
```

Assign execute rights to the installer and run the `django_fobi_example_app_installer.sh`:

```
chmod +x django_fobi_example_app_installer.sh
./django_fobi_example_app_installer.sh
```

Open your browser and test the app.

Dashboard:

- URL: <http://127.0.0.1:8001/fobi/>
- Admin username: test_admin
- Admin password: test

Django admin interface:

- URL: <http://127.0.0.1:8001/admin/>
- Admin username: test_admin
- Admin password: test

If quick installer doesn't work for you, see the manual steps on running the [example project](#).

QUICK START

See the [quick start](#).

INSTALLATION

- (1) Install latest stable version from PyPI:

```
pip install django-fobi
```

Or latest stable version from GitHub:

```
pip install https://github.com/barseghyanartur/django-fobi/archive/stable.tar.gz
```

- (2) Add *fobi* to `INSTALLED_APPS` of the your projects' Django settings. Furthermore, all themes and plugins to be used, shall be added to the `INSTALLED_APPS` as well. Note, that if a plugin has additional dependencies, you should be mentioning those in the `INSTALLED_APPS` as well.

```
INSTALLED_APPS = (  
    # Used by fobi  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.sites',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'django.contrib.admin',  
  
    # ...  
    # `django-fobi` core  
    'fobi',  
  
    # `django-fobi` themes  
    'fobi.contrib.themes.bootstrap3', # Bootstrap 3 theme  
    'fobi.contrib.themes.foundation5', # Foundation 5 theme  
    'fobi.contrib.themes.simple', # Simple theme  
  
    # `django-fobi` form elements - fields  
    'fobi.contrib.plugins.form_elements.fields.boolean',  
    'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple',  
    'fobi.contrib.plugins.form_elements.fields.date',  
    'fobi.contrib.plugins.form_elements.fields.date_drop_down',  
    'fobi.contrib.plugins.form_elements.fields.datetime',  
    'fobi.contrib.plugins.form_elements.fields.decimal',  
    'fobi.contrib.plugins.form_elements.fields.email',  
    'fobi.contrib.plugins.form_elements.fields.file',  
    'fobi.contrib.plugins.form_elements.fields.float',
```

(continues on next page)

(continued from previous page)

```

'fobi.contrib.plugins.form_elements.fields.hidden',
'fobi.contrib.plugins.form_elements.fields.input',
'fobi.contrib.plugins.form_elements.fields.integer',
'fobi.contrib.plugins.form_elements.fields.ip_address',
'fobi.contrib.plugins.form_elements.fields.null_boolean',
'fobi.contrib.plugins.form_elements.fields.password',
'fobi.contrib.plugins.form_elements.fields.radio',
'fobi.contrib.plugins.form_elements.fields.regex',
'fobi.contrib.plugins.form_elements.fields.select',
'fobi.contrib.plugins.form_elements.fields.select_model_object',
'fobi.contrib.plugins.form_elements.fields.select_multiple',
'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects',
'fobi.contrib.plugins.form_elements.fields.slug',
'fobi.contrib.plugins.form_elements.fields.text',
'fobi.contrib.plugins.form_elements.fields.textarea',
'fobi.contrib.plugins.form_elements.fields.time',
'fobi.contrib.plugins.form_elements.fields.url',

# `django-fobi` form elements - content elements
'fobi.contrib.plugins.form_elements.test.dummy',
'easy_thumbnails', # Required by `content_image` plugin
'fobi.contrib.plugins.form_elements.content.content_image',
'fobi.contrib.plugins.form_elements.content.content_image_url',
'fobi.contrib.plugins.form_elements.content.content_text',
'fobi.contrib.plugins.form_elements.content.content_video',

# `django-fobi` form handlers
'fobi.contrib.plugins.form_handlers.db_store',
'fobi.contrib.plugins.form_handlers.http_repost',
'fobi.contrib.plugins.form_handlers.mail',
'fobi.contrib.plugins.form_handlers.mail_sender',

# Other project specific apps
'foo', # Test app
# ...
)

```

- (3) Make appropriate changes to the `TEMPLATES` of the your projects' Django settings.

And `fobi.context_processors.theme` and `fobi.context_processors.dynamic_values`. See the following example.

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [(os.path.join('path', 'to', 'your', 'templates'))],
        'OPTIONS': {
            'context_processors': [
                "django.template.context_processors.debug",
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
                "fobi.context_processors.theme", # Important!
            ]
        }
    }
]

```

(continues on next page)

(continued from previous page)

```
        "fobi.context_processors.dynamic_values", # Optional
    ],
    'loaders': [
        'django.template.loaders.filesystem.Loader',
        'django.template.loaders.app_directories.Loader',
        'admin_tools.template_loaders.Loader',
    ],
    'debug': DEBUG_TEMPLATE,
}
},
]
```

Make sure that `django.core.context_processors.request` is in `context_processors` too.

(4) Configure URLs

Add the following line to urlpatterns of your *urls* module.

```
# View URLs
url(r'^fobi/', include('fobi.urls.view')),

# Edit URLs
url(r'^fobi/', include('fobi.urls.edit')),
```

Note, that some plugins require additional URL includes. For instance, if you listed the `fobi.contrib.plugins.form_handlers.db_store` form handler plugin in the `INSTALLED_APPS`, you should mention the following in `urls` module.

```
# DB Store plugin URLs
url(r'^fobi/plugins/form-handlers/db-store/',
    include('fobi.contrib.plugins.form_handlers.db_store.urls')),
```

View URLs are put separately from edit URLs in order to make it possible to prefix the edit URLs differently. For example, if you’re using the “Simple” theme, you would likely want to prefix the edit URLs with “admin/” so that it looks more like django-admin.

CREATING A NEW FORM ELEMENT PLUGIN

Form element plugins represent the elements of which the forms is made: Inputs, checkboxes, textareas, files, hidden fields, as well as pure presentational elements (text or image). Number of form elements in a form is not limited.

Presentational form elements are inherited from `fobi.base.FormElementPlugin`.

The rest (real form elements, that are supposed to have a value) are inherited from `fobi.base.FormFieldPlugin`.

You should see a form element plugin as a Django micro app, which could have its' own models, admin interface, etc.

django-fobi comes with several bundled form element plugins. Do check the source code as example.

Let's say, you want to create a textarea form element plugin.

There are several properties, each textarea should have. They are:

- *label* (string): HTML label of the textarea.
- *name* (string): HTML name of the textarea.
- *initial* (string): Initial value of the textarea.
- *required* (bool): Flag, which tells us whether the field is required or optional.

Let's name that plugin `sample_textarea`. The plugin directory should then have the following structure.

```
path/to/sample_textarea/  
├── __init__.py  
├── fobi_form_elements.py # Where plugins are defined and registered  
├── forms.py # Plugin configuration form  
└── widgets.py # Where plugins widgets are defined
```

Form element plugins should be registered in “`fobi_form_elements.py`” file. Each plugin module should be put into the `INSTALLED_APPS` of your Django projects' settings.

In some cases, you would need plugin specific overridable settings (see `fobi.contrib.form_elements.fields.content.content_image` plugin as an example). You are advised to write your settings in such a way, that variables of your Django project settings module would have *FOBI_PLUGIN_* prefix.

9.1 Define and register the form element plugin

Step by step review of a how to create and register a plugin and plugin widgets. Note, that *django-fobi* auto-discovers your plugins if you place them into a file named `fobi_form_elements.py` of any Django app listed in `INSTALLED_APPS` of your Django projects' settings module.

9.1.1 `path/to/sample_textarea/fobi_form_elements.py`

A single form element plugin is registered by its' UID.

Required imports.

```
from django import forms
from fobi.base import FormFieldPlugin, form_element_plugin_registry
from path.to.sample_textarea.forms import SampleTextareaForm
```

Defining the Sample textarea plugin.

```
class SampleTextareaPlugin(FormFieldPlugin):
    """Sample textarea plugin."""

    uid = "sample_textarea"
    name = "Sample Textarea"
    form = SampleTextareaForm
    group = "Samples" # Group to which the plugin belongs to

    def get_form_field_instances(self,
                                request=None,
                                form_entry=None,
                                form_element_entries=None,
                                **kwargs):

        kwargs = {
            'required': self.data.required,
            'label': self.data.label,
            'initial': self.data.initial,
            'widget': forms.widgets.Textarea(attrs={})
        }

        return [(self.data.name, forms.CharField, kwargs),]
```

Registering the SampleTextareaPlugin plugin.

```
form_element_plugin_registry.register(SampleTextareaPlugin)
```

Note, that in case you want to define a pure presentational element, make use of `fobi.base.FormElementPlugin` for subclassing, instead of `fobi.base.FormFieldPlugin`. See the source of the content plugins (`fobi.contrib.plugins.form_elements.content`) as an example.

For instance, the `captcha` and `honeypot` fields are implemented as form elements (subclasses the `fobi.base.FormElementPlugin`). The `db_store` form handler plugin does not save the form data of those elements. If you want the form element data to be saved, do inherit from `fobi.base.FormFieldPlugin`.

Hidden form element plugins, should be also having set the `is_hidden` property to `True`. By default it's set to `False`. That makes the hidden form elements to be rendered using as `django.forms.widgets.TextInput` widget in edit mode. In the view mode, the original widget that you assigned in your form element plugin would be used.

There might be cases, when you need to do additional handling of the data upon the successful form submission. In such cases, you will need to define a `submit_plugin_form_data` method in the plugin, which accepts the following arguments:

- `form_entry` (`fobi.models.FormEntry`): Form entry, which is being submitted.
- `request` (`django.http.HttpRequest`): The Django HTTP request.
- `form` (`django.forms.Form`): Form object (a valid one, which contains the `cleaned_data` attribute).
- `form_element_entries` (`fobi.models.FormElementEntry`): Form element entries for the `form_entry` given.
- `(**kwargs)` : Additional arguments.

Example (taken from `fobi.contrib.plugins.form_elements.fields.file`):

```
def submit_plugin_form_data(self,
                            form_entry,
                            request,
                            form,
                            form_element_entries=None,
                            **kwargs):
    """Submit plugin form data."""
    # Get the file path
    file_path = form.cleaned_data.get(self.data.name, None)
    if file_path:
        # Handle the upload
        saved_file = handle_uploaded_file(FILE_UPLOAD_DIR, file_path)
        # Overwrite ``cleaned_data`` of the ``form`` with path to moved
        # file.
        form.cleaned_data[self.data.name] = "{0}{1}".format(
            settings.MEDIA_URL, saved_file
        )

    # It's critically important to return the ``form`` with updated
    # ``cleaned_data``
    return form
```

In the example below, the original form is being modified. If you don't want the original form to be modified, do not return anything.

Check the file form element plugin (`fobi.contrib.plugins.form_elements.fields.file`) for complete example.

9.1.2 path/to/sample_textarea/forms.py

Why to have another file for defining forms? Just to keep the code clean and less messy, although you could perfectly define all your plugin forms in the module `fobi_form_elements.py`, it's recommended to keep it separate.

Take into consideration, that `forms.py` is not an auto-discovered file pattern. All your form element plugins should be registered in modules named `fobi_form_elements.py`.

Required imports.

```
from django import forms
from fobi.base import BasePluginForm
```

Form for for `SampleTextareaPlugin` form element plugin.

```
class SampleTextareaForm(forms.Form, BasePluginForm):
    """Sample textarea form."""

    plugin_data_fields = [
        ("name", ""),
        ("label", ""),
        ("initial", ""),
        ("required", False)
    ]

    name = forms.CharField(label="Name", required=True)
    label = forms.CharField(label="Label", required=True)
    initial = forms.CharField(label="Initial", required=False)
    required = forms.BooleanField(label="Required", required=False)
```

Note that although it's not being checked in the code, but for form field plugins the following fields should be present in the plugin form (BasePluginForm) and the form plugin (FormFieldPlugin):

- name

In some cases, you might want to do something with the data before it gets saved. For that purpose, `save_plugin_data` method has been introduced.

See the following [example](#).

```
def save_plugin_data(self, request=None):
    """Saving the plugin data and moving the file."""
    file_path = self.cleaned_data.get('file', None)
    if file_path:
        saved_image = handle_uploaded_file(IMAGES_UPLOAD_DIR, file_path)
        self.cleaned_data['file'] = saved_image
```

9.1.3 path/to/sample_textarea/widgets.py

Required imports.

```
from fobi.base import FormElementPluginWidget
```

Defining the base plugin widget.

```
class BaseSampleTextareaPluginWidget(FormElementPluginWidget):
    """Base sample textarea plugin widget."""

    # Same as `uid` value of the `SampleTextareaPlugin`.
    plugin_uid = "sample_textarea"
```


9.1.4 path/to/sample_layout/fobi_form_elements.py

Register in the registry (in some module which is for sure to be loaded; it's handy to do it in the theme module).

Required imports.

```
from fobi.base import form_element_plugin_widget_registry
from path.to.sample_textarea.widgets import BaseSampleTextareaPluginWidget
```

Define the theme specific plugin.

```
class SampleTextareaPluginWidget(BaseSampleTextareaPluginWidget):
    """Sample textarea plugin widget."""

    theme_uid = 'bootstrap3' # Theme for which the widget is loaded
    media_js = [
        'sample_layout/js/fobi.plugins.form_elements.sample_textarea.js',
    ]
    media_css = [
        'sample_layout/css/fobi.plugins.form_elements.sample_textarea.css',
    ]
```

Register the widget.

```
form_element_plugin_widget_registry.register(SampleTextareaPluginWidget)
```

9.1.5 Form element plugin final steps

Now, that everything is ready, make sure your plugin module is added to `INSTALLED_APPS`.

```
INSTALLED_APPS = (
    # ...
    'path.to.sample_textarea',
    # ...
)
```

Afterwards, go to terminal and type the following command.

```
./manage.py fobi_sync_plugins
```

If your HTTP server is running, you would then be able to see the new plugin in the edit form interface.

Dashboard URL: <http://127.0.0.1:8000/fobi/>

Note, that you have to be logged in, in order to use the dashboard. If your new plugin doesn't appear, set the `FOBI_DEBUG` to `True` in your Django's local settings module, re-run your code and check console for error notifications.

CREATING A NEW FORM HANDLER PLUGIN

Form handler plugins handle the form data. *django-fobi* comes with several bundled form handler plugins, among which is the `db_store` and `mail` plugins, which are responsible for saving the submitted form data into the database and mailing the data to recipients specified. Number of form handlers in a form is not limited. Certain form handlers are not configurable (for example the `db_store` form handler isn't), while others are (`mail`, `http_repost`).

You should see a form handler as a Django micro app, which could have its' own models, admin interface, etc.

By default, it's possible to use a form handler plugin multiple times per form. If you wish to allow form handler plugin to be used only once in a form, set the `allow_multiple` property of the plugin to `False`.

As said above, *django-fobi* comes with several bundled form handler plugins. Do check the source code as example.

10.1 Define and register the form handler plugin

Let's name that plugin `sample_mail`. The plugin directory should then have the following structure.

```
path/to/sample_mail/
├── __init__.py
├── fobi_form_handlers.py # Where plugins are defined and registered
└── forms.py # Plugin configuration form
```

Form handler plugins should be registered in “`fobi_form_handlers.py`” file. Each plugin module should be put into the `INSTALLED_APPS` of your Django projects' settings.

10.1.1 path/to/sample_mail/fobi_form_handlers.py

A single form handler plugin is registered by its' UID.

Required imports.

```
import json
from django.core.mail import send_mail
from fobi.base import FormHandlerPlugin, form_handler_plugin_registry
from path.to.sample_mail.forms import SampleMailForm
```

Defining the Sample mail handler plugin.

```
class SampleMailHandlerPlugin(FormHandlerPlugin):
    """Sample mail handler plugin."""
```

(continues on next page)

(continued from previous page)

```

uid = "sample_mail"
name = _("Sample mail")
form = SampleMailForm

def run(self, form_entry, request, form, form_element_entries=None):
    """To be executed by handler."""
    send_mail(
        self.data.subject,
        json.dumps(form.cleaned_data),
        self.data.from_email,
        [self.data.to_email],
        fail_silently=True
    )

```

Register the plugin

```
form_handler_plugin_registry.register(SampleMailHandlerPlugin)
```

Some form handlers are configurable, some others not. In order to have a user friendly way of showing the form handler settings, what's sometimes needed, a `plugin_data_repr` method has been introduced. Simplest implementation of it would look as follows:

```

def plugin_data_repr(self):
    """Human readable representation of plugin data.

    :return string:
    """
    return self.data.__dict__

```

10.1.2 path/to/sample_mail/forms.py

If plugin is configurable, it has configuration data. A single form may have unlimited number of same plugins. Imagine, you want to have different subjects and additional body texts for different user groups. You could then assign two form handler mail plugins to the form. Of course, saving the posted form data many times does not make sense, but it's up to the user. So, in case if plugin is configurable, it should have a form.

Why to have another file for defining forms? Just to keep the code clean and less messy, although you could perfectly define all your plugin forms in the module `fobi_form_handlers.py`, it's recommended to keep it separate.

Take into consideration, that `forms.py` is not an auto-discovered file pattern. All your form handler plugins should be registered in modules named `fobi_form_handlers.py`.

Required imports.

```

from django import forms
from django.utils.translation import gettext_lazy as _
from fobi.base import BasePluginForm

```

Defining the form for Sample mail handler plugin.

```

class MailForm(forms.Form, BasePluginForm):
    """Mail form."""

```

(continues on next page)

(continued from previous page)

```

plugin_data_fields = [
    ("from_name", ""),
    ("from_email", ""),
    ("to_name", ""),
    ("to_email", ""),
    ("subject", ""),
    ("body", ""),
]

from_name = forms.CharField(label=_("From name"), required=True)
from_email = forms.EmailField(label=_("From email"), required=True)
to_name = forms.CharField(label=_("To name"), required=True)
to_email = forms.EmailField(label=_("To email"), required=True)
subject = forms.CharField(label=_("Subject"), required=True)
body = forms.CharField(
    label=_("Body"),
    required=False,
    widget=forms.widgets.Textarea
)

```

After the plugin has been processed, all its' data is available in a `plugin_instance.data` container (for example, `plugin_instance.data.subject` or `plugin_instance.data.from_name`).

10.1.3 Prioritise the execution order

Some form handlers shall be executed prior others. A good example of such, is a combination of “mail” and “db_save” form handlers for the form. In case if large files are posted, submission of form data would fail if “mail” plugin would be executed after “db_save” has been executed. That’s why it’s possible to prioritise that ordering in a `FOBI_FORM_HANDLER_PLUGINS_EXECUTION_ORDER` setting variable.

If not specified or left empty, form handler plugins would be ran in the order of discovery. All form handler plugins that are not listed in the `FORM_HANDLER_PLUGINS_EXECUTION_ORDER`, would be ran after the plugins that are mentioned there.

```

FORM_HANDLER_PLUGINS_EXECUTION_ORDER = (
    'http_repost',
    'mail',
    # The 'db_store' is left out intentionally, since it should
    # be the last plugin to be executed.
)

```

10.1.4 Form handler plugin custom actions

By default, a single form handler plugin has at least a “delete” action. If plugin is configurable, it gets an “edit” action as well.

For some of your plugins, you may want to register a custom action. For example, the “db_store” plugin does have one, for showing a link to a listing page with saved form data for the form given.

For such cases, define a `custom_actions` method in your form handler plugin. That method shall return a list of triples. In each triple, first value is the URL, second value is the title and the third value is the icon of the URL.

The following example is taken from the “db_store” plugin.

```
def custom_actions(self):
    """Adding a link to view the saved form entries.

    :return iterable:
    """
    return (
        (
            reverse('fobi.contrib.plugins.form_handlers.db_store.view_saved_form_data_
↪entries'),
            _("View entries"),
            'glyphicon glyphicon-list'
        ),
    )
```

10.1.5 Form handler plugin final steps

Do not forget to add the form handler plugin module to `INSTALLED_APPS`.

```
INSTALLED_APPS = (
    # ...
    'path.to.sample_mail',
    # ...
)
```

Afterwards, go to terminal and type the following command.

```
./manage.py fobi_sync_plugins
```

If your HTTP server is running, you would then be able to see the new plugin in the edit form interface.

CREATING A NEW FORM IMPORTER PLUGIN

Form importer plugins import the forms from some external data source into *django-fobi* form format. Number of form importers is not limited. Form importers are implemented in forms of wizards (since they may contain several steps).

You should see a form importer as a Django micro app, which could have its' own models, admin interface, etc.

At the moment *django-fobi* comes with only one bundled form handler plugin, which is the `mailchimp_importer`, which is responsible for importing existing MailChimp forms into *django-fobi*.

11.1 Define and register the form importer plugin

Let's name that plugin `sample_importer`. The plugin directory should then have the following structure.

```
path/to/sample_importer/
├── templates
│   └── sample_importer
│       ├── 0.html
│       └── 1.html
├── __init__.py
├── fobi_form_importers.py # Where plugins are defined and registered
├── forms.py # Wizard forms
└── views.py # Wizard views
```

Form importer plugins should be registered in “`fobi_form_importers.py`” file. Each plugin module should be put into the `INSTALLED_APPS` of your Django projects' settings.

11.1.1 path/to/sample_importer/fobi_form_importers.py

A single form importer plugin is registered by its' UID.

Required imports.

```
from django.utils.translation import gettext_lazy as _
from fobi.form_importers import BaseFormImporter, form_importer_plugin_registry
from fobi.contrib.plugins.form_elements import fields
from path.to.sample_importer.views import SampleImporterWizardView
```

Defining the Sample importer plugin.

```
class SampleImporterPlugin(FormHandlerPlugin):
    """Sample importer plugin."""

    uid = 'sample_importer'
    name = _("Sample importer")
    wizard = SampleImporterWizardView
    templates = [
        'sample_importer/0.html',
        'sample_importer/1.html',
    ]

    # field_type (at importer): uid (django-fobi)
    fields_mapping = {
        # Implemented
        'email': fields.email.UID,
        'text': fields.text.UID,
        'number': fields.integer.UID,
        'dropdown': fields.select.UID,
        'date': fields.date.UID,
        'url': fields.url.UID,
        'radio': fields.radio.UID,

        # Transformed into something else
        'address': fields.text.UID,
        'zip': fields.text.UID,
        'phone': fields.text.UID,
    }

    # Django standard: remote
    field_properties_mapping = {
        'label': 'name',
        'name': 'tag',
        'help_text': 'helptext',
        'initial': 'default',
        'required': 'req',
        'choices': 'choices',
    }

    field_type_prop_name = 'field_type'
    position_prop_name = 'order'

    def extract_field_properties(self, field_data):
        field_properties = {}
        for prop, val in self.field_properties_mapping.items():
            if val in field_data:
                if 'choices' == val:
                    field_properties[prop] = "\n".join(field_data[val])
                else:
                    field_properties[prop] = field_data[val]
        return field_properties

form_importer_plugin_registry.register(SampleImporter)
```


11.1.2 path/to/sample_importer/forms.py

As mentioned above, form importers are implemented in form of wizards. The forms are the wizard steps.

Required imports.

```
from django import forms
from django.utils.translation import gettext_lazy as _
from sample_service_api import sample_api # Just an imaginary API client
```

Defining the form for Sample importer plugin.

```
class SampleImporterStep1Form(forms.Form):
    """First form the the wizard."""

    api_key = forms.CharField(required=True)

class SampleImporterStep2Form(forms.Form):
    """Second form of the wizard."""

    list_id = forms.ChoiceField(required=True, choices=[])

    def __init__(self, *args, **kwargs):
        self._api_key = None

        if 'api_key' in kwargs:
            self._api_key = kwargs.pop('api_key', None)

        super(SampleImporterStep2Form, self).__init__(*args, **kwargs)

        if self._api_key:
            client = sample_api.Api(self._api_key)
            lists = client.lists.list()
            choices = [(l['id'], l['name']) for l in lists['data']]
            self.fields['list_id'].choices = choices
```

11.1.3 path/to/sample_importer/views.py

The wizard views.

Required imports.

```
from sample_service_api import sample_api # Just an imaginary API client

from django.shortcuts import redirect
from django.urls import reverse
from django.contrib import messages
from django.utils.translation import gettext_lazy as _

# For django LTE 1.8 import from `django.contrib.formtools.wizard.views`
from formtools.wizard.views import SessionWizardView
```

(continues on next page)

(continued from previous page)

```
from path.to.sample_importer.forms import (
    SampleImporterStep1Form,
    SampleImporterStep2Form,
)
```

Defining the wizard view for Sample importer plugin.

```
class SampleImporterWizardView(SessionWizardView):
    """Sample importer wizard view."""

    form_list = [SampleImporterStep1Form, SampleImporterStep2Form]

    def get_form_kwargs(self, step):
        """Get form kwargs (to be used internally)."""
        if '1' == step:
            data = self.get_cleaned_data_for_step('0') or {}
            api_key = data.get('api_key', None)
            return {'api_key': api_key}
        return {}

    def done(self, form_list, **kwargs):
        """After all forms are submitted."""
        # Merging cleaned data into one dict
        cleaned_data = {}
        for form in form_list:
            cleaned_data.update(form.cleaned_data)

        # Connecting to sample client API
        client = sample_client.Api(cleaned_data['api_key'])

        # Fetching the form data
        form_data = client.lists.merge_vars(
            id={'list_id': cleaned_data['list_id']}
        )

        # We need the first form only
        try:
            form_data = form_data['data'][0]
        except Exception as err:
            messages.warning(
                self.request,
                _('Selected form could not be imported due errors.')
            )
            return redirect(reverse('fobi.dashboard'))

        # Actually, import the form
        form_entry = self._form_importer.import_data(
            {'name': form_data['name'], 'user': self.request.user},
            form_data['merge_vars']
        )

        redirect_url = reverse(
```

(continues on next page)

(continued from previous page)

```
        'fobi.edit_form_entry',
        kwargs={'form_entry_id': form_entry.pk}
    )

    messages.info(
        self.request,
        _('Form {0} imported successfully.').format(form_data['name'])
    )

    return redirect("{0}".format(redirect_url))
```

11.1.4 Form importer plugin final steps

Do not forget to add the form importer plugin module to `INSTALLED_APPS`.

```
INSTALLED_APPS = (
    # ...
    'path.to.sample_importer',
    # ...
)
```

Afterwards, go to terminal and type the following command.

```
./manage.py fobi_sync_plugins
```

If your HTTP server is running, you would then be able to see the new plugin in the dashboard form interface (implemented in all bundled themes).

CREATING A FORM CALLBACK

Form callbacks are additional hooks, that are executed on various stages of the form submission.

Let's place the callback in the `foo` module. The plugin directory should then have the following structure.

```
path/to/foo/  
├── __init__.py  
└── fobi_form_callbacks.py # Where callbacks are defined and registered
```

See the callback example below.

Required imports.

```
from fobi.constants import (  
    CALLBACK_BEFORE_FORM_VALIDATION,  
    CALLBACK_FORM_VALID_BEFORE_SUBMIT_PLUGIN_FORM_DATA,  
    CALLBACK_FORM_VALID, CALLBACK_FORM_VALID_AFTER_FORM_HANDLERS,  
    CALLBACK_FORM_INVALID  
)  
from fobi.base import FormCallback, form_callback_registry
```

Define and register the callback

```
class SampleFooCallback(FormCallback):  
    """Sample foo callback."""  
  
    stage = CALLBACK_FORM_VALID  
  
    def callback(self, form_entry, request, form):  
        """Define your callback code here."""  
        print("Great! Your form is valid!")  
  
form_callback_registry.register(SampleFooCallback)
```

Add the callback module to `INSTALLED_APPS`.

```
INSTALLED_APPS = (  
    # ...  
    'path.to.foo',  
    # ...  
)
```


CLASS-BASED VIEWS

13.1 Views

Migration to class based views is simple. Only your project's `urls.py` would change:

```
urlpatterns = [
    # ...
    url(r'^fobi/', include('fobi.urls.class_based.view')),
    url(r'^fobi/', include('fobi.urls.class_based.edit')),
    # ...
]
```

To use function based views, simply replace the previous line with:

```
urlpatterns = [
    # ...
    url(r'^fobi/', include('fobi.urls.view')),
    url(r'^fobi/', include('fobi.urls.edit')),
    # ...
]
```

13.2 Permissions

Class-based permissions work only in combination with class-based views.

Example:

```
from fobi.permissions.definitions import edit_form_entry_permissions
from fobi.permissions.generic import BasePermission
from fobi.permissions.helpers import (
    any_permission_required_func, login_required,
)

class EditFormEntryPermission(BasePermission):
    """Permission to edit form entries."""

    def has_permission(self, request, view) -> bool:
        return login_required(request) and any_permission_required_func(
            edit_form_entry_permissions
```

(continues on next page)

(continued from previous page)

```
) (request.user)

def has_object_permission(self, request, view, obj) -> bool:
    return login_required(request) and any_permission_required_func(
        edit_form_entry_permissions
    )(request.user) and obj.user == request.user
```

And then in your view:

```
from fobi.views.class_based import EditFormEntryView

class MyEditFormEntryView(EditFormEntryView):
    """EditFormEntryView."""

    permission_classes = (EditFormEntryPermission,)
```


SUGGESTIONS

14.1 Custom action for the form

Sometimes, you would want to specify a different action for the form. Although it's possible to define a custom form action (action field in the “Form properties” tab), you're advised to use the `http_repost` plugin instead, since then the form would be still validated locally and only then the valid data, as is, would be sent to the desired endpoint.

Take in mind, that if both cases, if CSRF protection is enabled on the endpoint, your post request would result an error.

14.2 When you want to customise too many things

django-fobi, with its' flexible form elements, form handlers and form callbacks is very customisable. However, there might be cases when you need to override entire view to fit your needs. Take a look at the [FeinCMS integration](#) or [DjangoCMS integration](#) as a good example of such. You may also want to compare the code from original view `fobi.views.view_form_entry` with the code from the widget to get a better idea of what could be changed in your case. If need a good advice, just ask me.

THEMING

django-fobi comes with theming API. While there are several ready-to-use themes:

- “Bootstrap 3” theme
- “Foundation 5” theme
- “Simple” theme in (with editing interface in style of the Django admin)
- “DjangoCMS admin style” theme (which is another simple theme with editing interface in style of `djangoCMS-admin-style`)

Obviously, there are two sorts of views when it comes to editing and viewing the form.

- The “view-view”, when the form as it has been made is exposed to the site end- users/visitors.
- The “edit-view” (builder view), where the authorised users build their forms.

Both “Bootstrap 3” and “Foundation 5” themes are making use of the same style for both “view-view” and “edit-view” views.

Both “Simple” and “DjangoCMS admin style” themes are styling for the “edit-view” only. The “view-view” is pretty much blank, as shown on the one of the screenshots [2.6].

Have in mind, that creating a brand new theme could be time consuming. Instead, you are advised to extend existing themes or in the worst case, if too much customisation required, create your own themes based on existing ones (just copy the desired theme to your project directory and work it out further).

It’s possible to use different templates for all “view” and “edit” actions (see the source code of the “simple” theme). Both “Bootstrap 3” and “Foundation 5” themes look great. Although if you can’t use any of those, the “Simple” theme is the best start, since it looks just like `django-admin`.

15.1 Create a new theme

Let’s place the theme in the `sample_theme` module. The theme directory should then have the following structure.

```
path/to/sample_theme/
├── static
│   ├── css
│   │   └── sample_theme.css
│   └── js
│       └── sample_theme.js
└── templates
    ├── sample_theme
    │   └── _base.html
```

(continues on next page)

(continued from previous page)

```

├── add_form_element_entry.html
├── ...
├── view_form_entry_ajax.html
├── __init__.py
├── fobi_form_elements.py
└── fobi_themes.py # Where themes are defined and registered

```

See the theme example below.

```

from django.utils.translation import gettext_lazy as _

from fobi.base import BaseTheme, theme_registry

class SampleTheme(BaseTheme):
    """Sample theme."""

    uid = 'sample'
    name = _("Sample")

    media_css = (
        'sample_theme/css/sample_theme.css',
        'css/fobi.core.css',
    )

    media_js = (
        'js/jquery-1.10.2.min.js',
        'jquery-ui/js/jquery-ui-1.10.3.custom.min.js',
        'js/jquery.slugify.js',
        'js/fobi.core.js',
        'sample_theme/js/sample_theme.js',
    )

    # Form element specific
    form_element_html_class = 'form-control'
    form_radio_element_html_class = 'radio'
    form_element_checkbox_html_class = 'checkbox'

    form_edit_form_entry_option_class = 'glyphicon glyphicon-edit'
    form_delete_form_entry_option_class = 'glyphicon glyphicon-remove'
    form_list_container_class = 'list-inline'

    # Templates
    master_base_template = 'sample_theme/_base.html'
    base_template = 'sample_theme/base.html'

    form_ajax = 'sample_theme/snippets/form_ajax.html'
    form_snippet_template_name = 'sample_theme/snippets/form_snippet.html'
    form_properties_snippet_template_name = 'sample_theme/snippets/form_properties_
↪ snippet.html'
    messages_snippet_template_name = 'sample_theme/snippets/messages_snippet.html'

    add_form_element_entry_template = 'sample_theme/add_form_element_entry.html'

```

(continues on next page)

(continued from previous page)

```

add_form_element_entry_ajax_template = 'sample_theme/add_form_element_entry_ajax.html'
↪

add_form_handler_entry_template = 'sample_theme/add_form_handler_entry.html'
add_form_handler_entry_ajax_template = 'sample_theme/add_form_handler_entry_ajax.html'
↪

create_form_entry_template = 'sample_theme/create_form_entry.html'
create_form_entry_ajax_template = 'bootstrap3/create_form_entry_ajax.html'

dashboard_template = 'sample_theme/dashboard.html'

edit_form_element_entry_template = 'sample_theme/edit_form_element_entry.html'
edit_form_element_entry_ajax_template = 'sample_theme/edit_form_element_entry_ajax.
↪html'

edit_form_entry_template = 'sample_theme/edit_form_entry.html'
edit_form_entry_ajax_template = 'sample_theme/edit_form_entry_ajax.html'

edit_form_handler_entry_template = 'sample_theme/edit_form_handler_entry.html'
edit_form_handler_entry_ajax_template = 'sample_theme/edit_form_handler_entry_ajax.
↪html'

form_entry_submitted_template = 'sample_theme/form_entry_submitted.html'
form_entry_submitted_ajax_template = 'sample_theme/form_entry_submitted_ajax.html'

view_form_entry_template = 'sample_theme/view_form_entry.html'
view_form_entry_ajax_template = 'sample_theme/view_form_entry_ajax.html'

```

Registering the SampleTheme plugin.

```
theme_registry.register(SampleTheme)
```

Sometimes you would want to attach additional properties to the theme in order to use them later in templates (remember, current theme object is always available in templates under name `fobi_theme`).

For such cases you would need to define a variable in your project's settings module, called `FOBI_CUSTOM_THEME_DATA`. See the following code as example:

```

# `django-fobi` custom theme data for to be displayed in third party apps
# like `django-registrator`.
FOBI_CUSTOM_THEME_DATA = {
    'bootstrap3': {
        'page_header_html_class': '',
        'form_html_class': 'form-horizontal',
        'form_button_outer_wrapper_html_class': 'control-group',
        'form_button_wrapper_html_class': 'controls',
        'form_button_html_class': 'btn',
        'form_primary_button_html_class': 'btn-primary pull-right',
    },
    'foundation5': {
        'page_header_html_class': '',
        'form_html_class': 'form-horizontal',
    },
}

```

(continues on next page)

(continued from previous page)

```

        'form_button_outer_wrapper_html_class': 'control-group',
        'form_button_wrapper_html_class': 'controls',
        'form_button_html_class': 'radius button',
        'form_primary_button_html_class': 'btn-primary',
    },
    'simple': {
        'page_header_html_class': '',
        'form_html_class': 'form-horizontal',
        'form_button_outer_wrapper_html_class': 'control-group',
        'form_button_wrapper_html_class': 'submit-row',
        'form_button_html_class': 'btn',
        'form_primary_button_html_class': 'btn-primary',
    }
}

```

You would now be able to access the defined extra properties in templates as shown below.

```
<div class="{{ fobi_theme.custom_data.form_button_wrapper_html_class }}">
```

You likely would want to either remove the footer text or change it. Define a variable in your project's settings module, called `FOBI_THEME_FOOTER_TEXT`. See the following code as example:

```
FOBI_THEME_FOOTER_TEXT = gettext('&copy; django-fobi example site 2014')
```

Below follow the properties of the theme:

- `base_edit`
- `base_view`

There are generic templates made in order to simplify theming. Some of them you would never need to override. Some others, you would likely want to.

Templates that you likely would want to re-write in your custom theme implementation are marked with three asterisks (***):

```

generic
├── snippets
│   ├── form_ajax.html
│   ├── form_edit_ajax.html
│   ├── *** form_properties_snippet.html
│   ├── *** form_snippet.html
│   ├── --- form_edit_snippet.html (does not exist in generic templates)
│   ├── --- form_view_snippet.html (does not exist in generic templates)
│   ├── form_view_ajax.html
│   └── messages_snippet.html
├── _base.html
├── add_form_element_entry.html
├── add_form_element_entry_ajax.html
├── add_form_handler_entry.html
├── add_form_handler_entry_ajax.html
├── base.html
├── create_form_entry.html
└── create_form_entry_ajax.html

```

(continues on next page)

(continued from previous page)

```

├── *** dashboard.html
├── edit_form_element_entry.html
├── edit_form_element_entry_ajax.html
├── edit_form_entry.html
├── *** edit_form_entry_ajax.html
├── edit_form_handler_entry.html
├── edit_form_handler_entry_ajax.html
├── form_entry_submitted.html
├── *** form_entry_submitted_ajax.html
├── *** theme.html
├── view_form_entry.html
└── view_form_entry_ajax.html

```

From all of the templates listed above, the `_base.html` template is the most influenced by the Bootstrap 3 theme.

15.2 Make changes to an existing theme

As said above, making your own theme from scratch could be costly. Instead, you can override/reuse an existing one and change it to your needs with minimal efforts. See the [override simple theme](#) example. In order to see it in action, run the project with `settings_override_simple_theme` option:

```
./manage.py runserver --settings=settings_override_simple_theme
```

Details explained below.

15.2.1 Directory structure

```

override_simple_theme/
├── static
│   └── override_simple_theme
│       ├── css
│       │   └── override-simple-theme.css
│       └── js
│           └── override-simple-theme.js
├── templates
│   └── override_simple_theme
│       ├── snippets
│       │   └── form_ajax.html
│       └── base_view.html
├── __init__.py
└── fobi_themes.py # Where themes are defined and registered

```

15.2.2 fobi_themes.py

Overriding the “simple” theme.

```
__all__ = ('MySimpleTheme',)

from fobi.base import theme_registry

from fobi.contrib.themes.simple.fobi_themes import SimpleTheme

class MySimpleTheme(SimpleTheme):
    """My simple theme, inherited from `SimpleTheme` theme."""

    html_classes = ['my-simple-theme',]
    base_view_template = 'override_simple_theme/base_view.html'
    form_ajax = 'override_simple_theme/snippets/form_ajax.html'
```

Register the overridden theme. Note, that it's important to set the *force* argument to True, in order to override the original theme. Force can be applied only once (for an overridden element).

```
theme_registry.register(MySimpleTheme, force=True)
```

15.2.3 templates/override_simple_theme/base_view.html

```
{% extends "simple/base_view.html" %}

{% load static %}

{% block stylesheets %}
<link
  href="{% static 'override_simple_theme/css/override-simple-theme.css' %}"
  rel="stylesheet" media="all" />
{% endblock stylesheets %}

{% block main-wrapper %}
<div id="sidebar">
  <h2>It's easy to override a theme!</h2>
</div>

{{ block.super }}
{% endblock main-wrapper %}
```


15.2.4 templates/override_simple_theme/snippets/form_ajax.html

```
{% extends "fobi/generic/snippets/form_ajax.html" %}

{% block form_html_class %}basic-grey{% endblock %}
```


FORM WIZARDS

16.1 Basics

With form wizards you can split forms across multiple pages. State is maintained in one of the backends (at the moment the Session backend). Data processing is delayed until the submission of the final form.

In *django-fobi* wizards work in the following way:

- Number of forms in a form wizard is not limited.
- Form callbacks, handlers are totally ignored in form wizards. Instead, the form-wizard specific handlers (form wizard handlers) take over handling of the form data on the final step.

16.2 Bundled form wizard handler plugins

Below a short overview of the form wizard handler plugins. See the README.rst file in directory of each plugin for details.

- **DB store**: Stores form data in a database.
- **HTTP repost**: Repost the POST request to another endpoint.
- **Mail**: Send the form data by email.
- **Mail the sender**: Send the form data by email to the sender (submitter) of the form.

INTEGRATION WITH THIRD-PARTY APPS AND FRAMEWORKS

django-fobi has been successfully integrated into a number of diverse third-party apps and frameworks, such as: Django REST framework, Django CMS, FeinCMS, Mezzanine and Wagtail.

Certainly, integration into CMS is one case, integration into REST framework - totally another. In REST frameworks we no longer have forms as such. Context is very different. Handling of form data should obviously happen in a different way. Assembling of the form class isn't enough (in case of Django REST framework we assemble the serializer class).

In order to handle such level of integration, two additional sort of plugins have been introduced:

- `IntegrationFormElementPlugin`
- `IntegrationFormHandlerPlugin`

These plugins are in charge of representation of the form elements in a proper way for the package to be integrated and handling the submitted form data.

[Additional documentation](#) is available in the sub-package.

17.1 Sample *IntegrationFormElementPlugin*

Sample is taken from [here](#).

17.1.1 `base.py`

Define the form element plugin.

```
from django.utils.translation import gettext_lazy as _

from rest_framework.fields import EmailField

from fobi.base import IntegrationFormFieldPlugin
from fobi.contrib.apps.drf_integration import UID as INTEGRATE_WITH_UID
from fobi.contrib.apps.drf_integration.base import (
    DRFIntegrationFormElementPluginProcessor,
    DRFSubmitPluginFormDataMixin,
)
from fobi.contrib.apps.drf_integration.form_elements.fields.email import UID

class EmailInputPlugin(IntegrationFormFieldPlugin,
    DRFSubmitPluginFormDataMixin):
```

(continues on next page)

(continued from previous page)

```

"""EmailField plugin."""

uid = UID
integrate_with = INTEGRATE_WITH_UID
name = _("Decimal")
group = _("Fields")

def get_custom_field_instances(self,
                               form_element_plugin,
                               request=None,
                               form_entry=None,
                               form_element_entries=None,
                               **kwargs):
    """Get form field instances."""
    field_kwargs = {
        'required': form_element_plugin.data.required,
        'initial': form_element_plugin.data.initial,
        'label': form_element_plugin.data.label,
        'help_text': form_element_plugin.data.help_text,
        'max_length': form_element_plugin.data.max_length,
    }
    return [
        DRFIntegrationFormElementPluginProcessor(
            field_class=EmailField,
            field_kwargs=field_kwargs
        )
    ]

```

17.1.2 fobi_integration_form_elements.py

Register the plugin. Note the name pattern *fobi_integration_form_elements*.

```

from fobi.base import integration_form_element_plugin_registry
from .base import EmailInputPlugin

integration_form_element_plugin_registry.register(EmailInputPlugin)

```

Don't forget to list your plugin in the `INSTALLED_APPS` afterwards.

17.2 Sample *IntegrationFormHandlerPlugin*

Sample is taken from [here](#).

17.2.1 base.py

Define the form handler plugin.

```
import logging
from mimetypes import guess_type
import os

from django.conf import settings
from django.utils.translation import gettext_lazy as _

from fobi.base import IntegrationFormHandlerPlugin
from fobi.helpers import extract_file_path

from fobi.contrib.apps.drf_integration import UID as INTEGRATE_WITH_UID
from fobi.contrib.apps.drf_integration.base import get_processed_serializer_data

from . import UID


class MailHandlerPlugin(IntegrationFormHandlerPlugin):
    """Mail handler form handler plugin.

    Can be used only once per form.
    """

    uid = UID
    name = _("Mail")
    integrate_with = INTEGRATE_WITH_UID

    def run(self,
            form_handler_plugin,
            form_entry,
            request,
            form_element_entries=None,
            **kwargs):
        """Run."""
        base_url = form_handler_plugin.get_base_url(request)

        serializer = kwargs['serializer']

        # Clean up the values, leave our content fields and empty values.
        field_name_to_label_map, cleaned_data = get_processed_serializer_data(
            serializer,
            form_element_entries
        )

        rendered_data = form_handler_plugin.get_rendered_data(
```

(continues on next page)

(continued from previous page)

```

        serializer.validated_data,
        field_name_to_label_map,
        base_url
    )

    files = self._prepare_files(request, serializer)

    form_handler_plugin.send_email(rendered_data, files)

def _prepare_files(self, request, serializer):
    """Prepares the files for being attached to the mail message."""
    files = {}

    def process_path(file_path, imf):
        """Processes the file path and the file."""
        if file_path:
            file_path = file_path.replace(
                settings.MEDIA_URL,
                os.path.join(settings.MEDIA_ROOT, '')
            )
            mime_type = guess_type(imf.name)
            files[field_name] = (
                imf.name,
                ''.join([c for c in imf.chunks()]),
                mime_type[0] if mime_type else ''
            )

    for field_name, imf in request.FILES.items():
        try:
            file_path = serializer.validated_data.get(field_name, '')
            process_path(file_path, imf)
        except Exception as err:
            file_path = extract_file_path(imf.name)
            process_path(file_path, imf)

    return files

```

17.2.2 fobi_integration_form_handlers.py

Register the plugin. Note the name pattern *fobi_integration_form_handlers*.

```

from fobi.base import integration_form_handler_plugin_registry
from .base import MailHandlerPlugin

integration_form_handler_plugin_registry.register(MailHandlerPlugin)

```

Don't forget to list your plugin in the `INSTALLED_APPS` afterwards.

PERMISSIONS

Plugin system allows administrators to specify the access rights to every plugin. *django-fobi* permissions are based on Django Users and User Groups. Access rights are manageable via Django admin (“/admin/fobi/formelement/”, “/admin/fobi/formhandler/”). If user doesn’t have the rights to access plugin, it doesn’t appear on his form even if has been added to it (imagine, you have once granted the right to use the news plugin to all users, but later on decided to limit it to Staff members group only). Note, that superusers have access to all plugins.

Plugin access rights management interface in Django admin

`Plugin`	`Users`	`Groups`	
Text	John Doe	Form builder users	
Textarea		Form builder users	
File	Oscar, John Doe	Staff members	
URL		Form builder users	
Hidden		Form builder users	

MANAGEMENT COMMANDS

There are several management commands available.

- *fobi_find_broken_entries*. Find broken form element/handler entries that occur when some plugin which did exist in the system, no longer exists.
- *fobi_sync_plugins*. Should be ran each time a new plugin is being added to the *django-fobi*.
- *fobi_update_plugin_data*. A mechanism to update existing plugin data in case if it had become invalid after a change in a plugin. In order for it to work, each plugin should implement and `update` method, in which the data update happens.

TUNING

There are number of *django-fobi* settings you can override in the settings module of your Django project:

- *FOBI_RESTRICT_PLUGIN_ACCESS* (bool): If set to True, (Django) permission system for dash plugins is enabled. Defaults to True. Setting this to False makes all plugins available for all users.
- *FOBI_DEFAULT_THEME* (str): Active (default) theme UID. Defaults to “bootstrap3”.
- *FORM_HANDLER_PLUGINS_EXECUTION_ORDER* (list of tuples): Order in which the form handlers are executed. See the “Prioritise the execution order” section for details.

For tuning of specific contrib plugin, see the docs in the plugin directory.

BUNDLED PLUGINS AND THEMES

django-fobi ships with number of bundled form element- and form handler- plugins, as well as themes which are ready to be used as is.

21.1 Bundled form element plugins

Below a short overview of the form element plugins. See the README.rst file in directory of each plugin for details.

21.1.1 Fields

Fields marked with asterisk (*) fall under the definition of text elements. It's possible to provide *Dynamic initial values* for text elements.

- Boolean (checkbox)
- Checkbox select multiple
- Date
- Date drop down (year, month, day selection drop-downs)
- DateTime
- Decimal
- Duration
- Email*
- File
- Float
- Hidden*
- Hidden model object
- Input
- Integer
- IP address*
- Null boolean
- Password*
- Radio select (radio button)

- Range select
- Regex
- Select (drop-down)
- Select model object (drop-down)
- Select multiple (drop-down)
- Select multiple model objects (drop-down)
- Select multiple with max (drop-down)
- Slider
- Slug*
- Text*
- Textarea*
- Time
- URL*

21.1.2 Content/presentation

Content plugins are presentational plugins, that make your forms look more complete and content rich.

- **Content image:** Insert an image.
- **Content image URL:** Insert an image URL.
- **Content markdown:** Add markdown text.
- **Content richtext:** Add rich text (based on [django-ckeditor](#) package).
- **Content text:** Add text.
- **Content video:** Add an embed YouTube or Vimeo video.

21.1.3 Security

- **CAPTCHA:** CAPTCHA integration, requires [django-simple-captcha](#) package.
- **ReCAPTCHA:** CAPTCHA integration, requires [django-recaptcha](#) package.
- **Invisible ReCAPTCHA:** Google invisible reCAPTCHA integration, with no additional dependencies.
- **Honeypot:** Anti-spam honeypot field.

21.1.4 MPTT fields

- Select MPTT model object (drop-down)
- Select multiple MPTT model objects (drop-down)

21.1.5 Test

Test plugins are made for dev purposes only.

- [Dummy](#): Solely for dev purposes.

21.2 Bundled form handler plugins

Below a short overview of the form handler plugins. See the README.rst file in directory of each plugin for details.

- [DB store](#): Stores form data in a database.
- [HTTP repost](#): Repost the POST request to another endpoint.
- [Mail](#): Send the form data by email.
- [Mail the sender](#): Send the form data by email to the sender.

21.3 Bundled themes

Below a short overview of the themes. See the README.rst file in directory of each theme for details.

- [Bootstrap 3](#): Bootstrap 3 theme.
- [Foundation 5](#): Foundation 5 theme.
- [Simple](#): Basic theme with form editing is in a style of Django admin.
- [DjangoCMS admin style](#): Basic theme with form editing is in a style of `djangocms-admin-style`.

THIRD-PARTY PLUGINS AND THEMES

List of remarkable third-party plugins:

- `fobi-phonenum` - A Fobi PhoneNumber form field plugin. Makes use of the `phonenum-ber_field.formfields.PhoneNumberField` and `phonenum-ber_field.widgets.PhoneNumberPrefixWidget`.

HTML5 FIELDS

The following HTML5 fields are supported in corresponding bundled plugins:

- date
- datetime
- email
- max
- min
- number
- url
- placeholder
- type

With the `fobi.contrib.plugins.form_elements.fields.input` support for HTML5 fields is extended to the following fields:

- autocomplete
- autofocus
- list
- multiple
- pattern
- step

LOADING INITIAL DATA USING GET ARGUMENTS

It's possible to provide initial data for the form using the GET arguments.

In that case, along with the field values, you should be providing an additional argument named “fobi_initial_data”, which doesn't have to hold a value. For example, if your form contains of fields named “email” and “age” and you want to provide initial values for those using GET arguments, you should be constructing your URL to the form as follows:

http://127.0.0.1:8001/fobi/view/test-form/?fobi_initial_data&email=test@example.com&age=19

DYNAMIC INITIAL VALUES

It's possible to provide a dynamic initial value for any of the text elements. In order to do that, you should use the build-in context processor or make your own one. The only requirement is that you should store all values that should be exposed in the form as a dict for `fobi_dynamic_values` dictionary key. Beware, that passing the original request object might be unsafe in many ways. Currently, a stripped down version of the request object is being passed as a context variable.

```
TEMPLATES = [
    {
        # ...
        'OPTIONS': {
            # ...
            'context_processors': [
                # ...
                "fobi.context_processors.theme", # Important!
                "fobi.context_processors.dynamic_values", # Optional
            ]
        },
    },
]
```

```
def dynamic_values(request):
    return {
        'fobi_dynamic_values': {
            'request': StrippedRequest(request),
            'now': datetime.datetime.now(),
            'today': datetime.date.today(),
        }
    }
```

In your GUI, you should be referring to the initial values in the following way:

```
{{ request.path }} {{ now }} {{ today }}
```

Note, that you should not provide the `fobi_dynamic_values.` as a prefix. Currently, the following variables are available in the `fobi.context_processors.dynamic_values` context processor:

- `request`: Stripped `HttpRequest` object.
 - `request.path`: A string representing the full path to the requested page, not including the scheme or domain.

(continues on next page)

(continued from previous page)

- `request.get_full_path()`: Returns the path, plus an appended query string, if applicable.
- `request.is_secure()`: Returns True if the request is secure; that is, if it was made with HTTPS.
- `request.is_ajax()`: Returns True if the request was made via an XMLHttpRequest, by checking the HTTP_X_REQUESTED_WITH header for the string 'XMLHttpRequest'.
- `request.META`: A stripped down standard Python dictionary containing the available HTTP headers.
 - `HTTP_ACCEPT_ENCODING`: Acceptable encodings for the response.
 - `HTTP_ACCEPT_LANGUAGE`: Acceptable languages for the response.
 - `HTTP_HOST`: The HTTP Host header sent by the client.
 - `HTTP_REFERER`: The referring page, if any.
 - `HTTP_USER_AGENT`: The client's user-agent string.
 - `QUERY_STRING`: The query string, as a single (un-parsed) string.
 - `REMOTE_ADDR`: The IP address of the client.
- `request.user`: Authenticated user.
 - `request.user.email`:
 - `request.user.get_username()`: Returns the username for the user. Since the User model can be swapped out, you should use this method instead of referencing the username attribute directly.
 - `request.user.get_full_name()`: Returns the first_name plus the last_name, with a space in between.
 - `request.user.get_short_name()`: Returns the first_name.
 - `request.user.is_anonymous()`:
- `now`: `datetime.datetime.now()`
- `today`: `datetime.date.today()`

SUBMITTED FORM ELEMENT PLUGINS VALUES

While some values of form element plugins are submitted as is, some others need additional processing. There are 3 types of behaviour taken into consideration:

- “val”: value is being sent as is.
- “repr”: (human readable) representation of the value is used.
- “mix”: mix of value as is and human readable representation.

The following plugins have been made configurable in such a way, that developers can choose the desired behaviour in projects’ settings:

- `FOBI_FORM_ELEMENT_CHECKBOX_SELECT_MULTIPLE_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_RADIO_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_MODEL_OBJECT_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MODEL_OBJECTS_SUBMIT_VALUE_AS`

See the README.rst in each of the following plugins for more information.

- [Checkbox select multiple \(multiple checkboxes\)](#)
- [Radio select \(radio button\)](#)
- [Select \(drop-down\)](#)
- [Select model object \(drop-down\)](#)
- [Select MPTT model object \(drop-down\)](#)
- [Select multiple \(drop-down\)](#)
- [Select multiple model objects \(drop-down\)](#)
- [Select multiple MPTT model objects \(drop-down\)](#)

RENDERING FORMS USING THIRD-PARTY LIBRARIES

You might want to render your forms using third-party libraries such as [django-crispy-forms](#), [django-floppyforms](#) or other alternatives.

For that purpose you should override the “snippets/form_snippet.html” used by the theme you have chosen. Your template would then look similar to the one below (make sure to setup/configure your third-party form rendering library prior doing this).

27.1 Using *django-crispy-forms*

```
{% load crispy_forms_tags fobi_tags %}

{% block form_non_field_and_hidden_errors %}
    {% get_form_hidden_fields_errors form as form_hidden_fields_errors %}
    {% if form.non_field_errors or form_hidden_fields_errors %}
        {% include fobi_theme.form_non_field_and_hidden_errors_snippet_template %}
    {% endif %}
{% endblock form_non_field_and_hidden_errors %}

{% crispy form %}
```

27.2 Using *django-floppyforms*

```
{% load floppyforms fobi_tags %}

{% block form_non_field_and_hidden_errors %}
    {% get_form_hidden_fields_errors form as form_hidden_fields_errors %}
    {% if form.non_field_errors or form_hidden_fields_errors %}
        {% include fobi_theme.form_non_field_and_hidden_errors_snippet_template %}
    {% endif %}
{% endblock form_non_field_and_hidden_errors %}

{% form form %}
```

See how it's done in the [override simple theme](#) example.

IMPORT/EXPORT FORMS

There might be cases when you have *django-fobi* running on multiple instances and have already spend some time on making forms on one of the instances, and want to reuse those forms on another. You could of course re-create entire form in the GUI, but we can do better than that. It's possible to export forms into JSON format and import the exported forms again. It's preferable that you run both instances on the same versions of *django-fobi*, otherwise imports might break (although it might just work). There many ways to deal with missing plugin errors, but the chosen strategy (which you don't yet have full control of) is safest (import everything possible, but warn user about errors). If both instances have the same set of form element and form handler plugins imports should go smoothly. It is though possible to make an import ignoring missing form element and form handler plugins. You would get an appropriate notice about that, but import will continue leaving the broken plugin data out.

TRANSLATIONS

29.1 Available translations

English is the primary language. The following translations are available (core and plugins)

- Dutch
- German
- Russian
- French

29.2 Overriding translations

There might be cases when you want to override certain translations. It's easily achievable with introduction of custom locale paths in your project.

See the following as a good example of overriding some English plugin labels.

- [custom settings](#)
- [custom locales directory](#)

Run the example project as follows:

```
cd examples/simple/  
./manage.py runserver --settings=settings.alternative_labels
```

In the example given, “Boolean” and “Checkbox select multiple” plugin names are renamed to “Checkbox” and “Multiple checkboxes” respectively.

All built-in plugin name values are almost equivalent to the plugin uid values. By default plugins are sorted by uid value. When you override the name of the plugin, sorting breaks. Therefore, it's recommended to set the `FOBI_SORT_PLUGINS_BY_VALUE` value to `True` in your settings module. Default value is `False`, which means that plugins are sorted by their uid value.

```
FOBI_SORT_PLUGINS_BY_VALUE = True
```


DEBUGGING

By default debugging is turned off. It means that broken form entries, which are entries with broken data, that are not possible to be shown, are just skipped. That's safe in production. Although, you for sure would want to see the broken entries in development. Set the `FOBI_DEBUG` to `True` in the `settings.py` of your project in order to do so.

Most of the errors are logged (DEBUG). If you have written a plugin and it somehow doesn't appear in the list of available plugins, do run the following management command since it not only syncs your plugins into the database, but also is a great way of checking for possible errors.

```
./manage.py fobi_sync_plugins
```

Run the following command in order to identify the broken plugins.

```
./manage.py fobi_find_broken_entries
```

If you have forms referring to form element- of form handler- plugins that are currently missing (not registered, removed, failed to load - thus there would be a risk that your form wouldn't be rendered properly/fully and the necessary data handling wouldn't happen either) you will get an appropriate exception. Although it's fine to get an instant error message about such failures in development, in production it wouldn't look appropriate. Thus, there are two settings related to the non-existing (not-found) form element- and form handler- plugins.

- `FOBI_DEBUG`: Set this to `True` in your development environment anyway. Watch error logs closely.
- `FOBI_FAIL_ON_MISSING_FORM_ELEMENT_PLUGINS`: If you want no error to be shown in case of missing form element plugins, set this to `False` in your settings module. Default value is `True`.
- `FOBI_FAIL_ON_MISSING_FORM_HANDLER_PLUGINS`: If you want no error to be shown in case of missing form element handlers, set this to `False` in your settings module. Default value is `True`.

TESTING

Project is covered by test (functional- and browser-tests).

Note: You are recommended to use Postgres or MySQL for testing. Tests occasionally fail on SQLite due to very intensive IO and SQLite table locking.

To test with all supported Python/Django versions type:

```
tox
```

To test against specific environment, type:

```
tox -e py37-django21
```

To test just your working environment type:

```
./runtests.py
```

To run a single test class in a given test module in your working environment type:

```
./runtests.py src/fobi/tests/test_browser_build_dynamic_forms.  
↪py:FobiBrowserBuldDynamicFormsTest -k "test_2004_submit_form"
```

It's assumed that you have all the requirements installed. If not, first install the test requirements:

```
pip install -r examples/requirements/test.txt
```

31.1 Browser tests

For browser tests you may choose between Firefox, headless Firefox and PhantomJS. PhantomJS is faster, headless Firefox is fast as well, but normal Firefox tests tell you more (as you see what exactly happens on the screen). Both cases require some effort and both have disadvantages regarding the installation (although once you have them installed they work perfect).

Latest versions of Firefox are often not supported by Selenium. Current version of the Selenium for Python (2.53.6) works fine with Firefox 47. Thus, instead of using system Firefox you could better use a custom one.

For PhantomJS you need to have NodeJS installed.

31.1.1 Set up ChromeDriver

1. Download ChromeDriver version matching your browser. You can always find proper version at chromedriver.chromium.org/downloads:

```
wget https://chromedriver.storage.googleapis.com/87.0.4280.88/chromedriver_
linux64.zip
unzip chromedriver_linux64.zip
sudo mv chromedriver /usr/bin/chromedriver87
sudo chown root:root /usr/bin/chromedriver87
sudo chmod +x /usr/bin/chromedriver87
```

2. Specify the full path to your ChromeDriver in CHROME_DRIVER_EXECUTABLE_PATH setting. Example:

```
CHROME_DRIVER_EXECUTABLE_PATH = '/usr/bin/chromedriver87'
```

After that your Selenium tests would work.

31.1.2 Set up Firefox 47

1. Download Firefox 47 from [this](#) location and unzip it into /usr/lib/firefox47/
2. Specify the full path to your Firefox in FIREFOX_BIN_PATH setting. Example:

```
FIREFOX_BIN_PATH = '/usr/lib/firefox47/firefox'
```

If you set to use system Firefox, remove or comment-out the FIREFOX_BIN_PATH setting.

After that your Selenium tests would work.

31.1.3 Set up headless Firefox

1. Install xvfb package which is used to start Firefox in headless mode.

```
sudo apt-get install xvfb
```

2. Run the tests using headless Firefox.

```
./scripts/runtests.sh
```

Or run tox tests using headless Firefox.

```
./scripts/tox.sh
```

Or run specific tox tests using headless Firefox.

```
./scripts/tox.sh -e py36-django111
```

31.1.4 Setup PhantomJS

You could also run tests in headless mode (faster). For that you will need PhantomJS.

1. Install PhantomJS and dependencies.

```
curl -sL https://deb.nodesource.com/setup_6.x -o nodesource_setup.sh
sudo bash nodesource_setup.sh
sudo apt-get install nodejs
sudo apt-get install build-essential libssl-dev
sudo npm -g install phantomjs-prebuilt
```

2. Specify the PHANTOM_JS_EXECUTABLE_PATH setting. Example:

```
PHANTOM_JS_EXECUTABLE_PATH = ""
```

If you want to use Firefox for testing, remove or comment-out the PHANTOM_JS_EXECUTABLE_PATH setting.

WRITING DOCUMENTATION

Keep the following hierarchy.

```
====  
title  
====  
  
header  
=====  
  
sub-header  
-----  
  
sub-sub-header  
~~~~~  
  
sub-sub-sub-header  
#####  
  
sub-sub-sub-sub-header  
^^^^^^^^^^^^^^^^  
  
sub-sub-sub-sub-sub-header  
+++++
```


TROUBLESHOOTING

If you get a `FormElementPluginDoesNotExist` or a `FormHandlerPluginDoesNotExist` exception, make sure you have listed your plugin in the `settings` module of your project.

CONTRIBUTING

If you want to contribute to the library, but don't know where to start, do check the [open issues where help is appreciated](#) or ask the *Author* how you could help.

CHAPTER
THIRTYFIVE

LICENSE

GPL-2.0-only OR LGPL-2.1-or-later

SUPPORT

For any security issues contact me at the e-mail given in the *Author* section. For overall issues, go to [GitHub](#).

CHAPTER
THIRTYSEVEN

AUTHOR

Artur Barseghyan <artur.barseghyan@gmail.com>

SCREENSHOTS

38.1 Bootstrap3 theme

38.1.1 Dashboard

Build your forms

Dashboard

Your forms

Form	Is public	Is cloneable	Actions
Test form	True	False	Edit Delete
Test form 2	False	False	Edit Delete
Test form 3	False	False	Edit Delete
1st form	True	False	Edit Delete
2nd form	True	False	Edit Delete

Actions

[+ Create form](#)

38.1.2 Create a form

Build your forms

Create form

Fields marked with * are required

Name*

My test form

Public?



Makes your form visible to the public.

Success page title

Thank you!

Custom message title to display after valid form is submitted

Success page body

Vivamus pharetra dui in tincidunt dapibus! Mauris id congue tellus! Vestibulum massa felis, varius ut sem eu, sollicitudin lacinia enim. Nam quis turpis ut purus mattis mattis. Praesent ex orci, ultricies ac blandit quis, pellentesque ac magna. Aenean maximus sapien tortor! Etiam vel consectetur orci; in euismod elit! Cras ornare sollicitudin blandit. Sed at turpis non mauris pulvinar posuere sed.

Custom message text to display after valid form is submitted

Add

© django-fobi example site 2014

38.1.3 View/edit form

38.1.3.1 Form elements

Build your forms

Edit

View

Form My test form was created successfully.

Edit form

Elements

Handlers

Properties

Add elements to your form

Fields marked with * are required

Choose form element to add ▾

Save ordering

Build your forms

Edit

View

Edit form

Elements

Handlers

Properties

Add elements to your form

Fields marked with * are required

Choose form element to add ▾

Name (Text)*

[Edit](#) [Delete](#)

Description (Textarea)

[Edit](#) [Delete](#)

Content

Content video

Content text

Content image

Fields

Select multiple

Textarea

Select model object

URL

Text

DateTime

Radio

File

Boolean

Date

Integer

Hidden

Password

Email

Select

Testing

Dummy

ordering

© django-fobi example site 2014

Build your forms

Add "URL" element to the form

Fields marked with * are required

Label*

Name*

Help text

Initial

Max length*

255

Required

☐

Add

© django-fobi example site 2014

Build your forms

Edit

View

Edit form

Elements

Handlers

Properties

Add elements to your form

Fields marked with * are required

Choose form element to add ▾

Name (Text)*

 Edit  Delete

Description (Textarea)

 Edit  Delete

Website (URL)

 Edit  Delete

Save ordering

© django-fobi example site 2014

38.1.3.2 Form handlers

Build your forms
Edit
View

The form element plugin "URL" was added successfully.

Edit form

Elements
Handlers
Properties

Add handlers to your form

Choose form handler to add ▼

© django-fobi example site 2014

Build your forms

Edit

View

The form handler plugin "DB store" was added successfully.

Edit form

Elements

Handlers

Properties

Add handlers to your form

Handler	Actions
DB store	✕ Delete ☰ View entries

Choose form handler to add ▾

- DB store
- Mail
- HTTP Repost

© django-fobi example site 2014

Build your forms

Add "Mail" handler to the form

Fields marked with * are required

From name*	<input type="text" value="From name"/>
From email*	<input type="text" value="from@example.com"/>
To name*	<input type="text" value="To name"/>
To email*	<input type="text" value="to@example.com"/>
Subject*	<input type="text" value="My form #1 mail"/>
Body	<div>Donec nec consequat velit, id mollis massa! Sed dapibus vehicula eleifend. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Morbi placerat justo eu diam commodo semper. Ut dapibus cursus mi id molestie? Aliquam nisi velit, placerat hendrerit cursus rhoncus, malesuada id ante. Nam ultricies orci et velit hendrerit, dignissim accumsan nibh rhoncus. Integer metus.</div>

© django-fobi example site 2014

Build your forms

Edit

View

The form handler plugin "Mail" was added successfully.

Edit form

Elements

Handlers

Properties

Add handlers to your form

Choose form handler to add ▾

Handler

Actions

DB store

✕ Delete  View entries

Mail 

 Edit ✕ Delete

Build your forms

Edit

View

Edit form

Elements

Handlers

Properties

Form properties

Fields marked with * are required

Name*

My test form

☒ Public? ?

Success page title

Thank you!

Custom message title to display after valid form is submitted

Success page body

Vivamus pharetra dui in tincidunt dapibus! Mauris id congue tellus! Vestibulum massa felis, varius ut sem eu, sollicitudin lacinia enim. Nam quis turpis ut purus mattis mattis. Praesent ex orci, ultricies ac blandit quis, pellentesque ac magna. Aenean maximus sapien tortor! Etiam vel consectetur orci; in euismod elit! Cras ornare sollicitudin blandit. Sed at turpis non mauris pulvinar posuere sed.

Custom message text to display after valid form is submitted

Submit changes

Build your forms

Edit

View

View form

Fields marked with * are required

Name*

n.n.

Description

Interdum et malesuada fames ac ante ipsum primis in faucibus. Donec a tortor orci. Integer semper ligula quis tristique bibendum. Sed iaculis vehicula libero, vitae auctor justo efficitur quis. Curabitur et nunc dignissim ante fermentum consequat. Morbi nulla purus, pulvinar sed ornare at, maximus fringilla mauris. Pellentesque eu blandit purus. Aenean vehicula tempus orci, vel cursus neque metus.

Website

<http://delusionalinsanity.com/portfolio/>

Submit

© django-fobi example site 2014

Build your forms

Form My test form was submitted successfully.

Thank you!

Vivamus pharetra dui in tincidunt dapibus! Mauris id congue tellus! Vestibulum massa felis, varius ut sem eu, sollicitudin lacinia enim. Nam quis turpis ut purus mattis mattis. Praesent ex orci, ultricies ac blandit quis, pellentesque ac magna. Aenean maximus sapien tortor! Etiam vel consectetur orci; in euismod elit! Cras ornare sollicitudin blandit. Sed at turpis non mauris pulvinar posuere sed.

© django-fobi example site 2014

Build your forms

Add "Content video" element to the form

Fields marked with * are required

Title*	<input type="text" value="Delusional Insanity"/>
URL*	<input type="text" value="http://youtu.be/8GVlui0JK0M?list=UURKM8j-UZfo8FYY9S3GsfWw"/>
Size	<input type="text" value="500x400"/>

Add

© django-fobi example site 2014

Build your forms

Add "Boolean" element to the form

Fields marked with * are required

Label*

Name*

Help text

Initial ☐

Required ☐

© django-fobi example site 2014

Build your forms
Edit
View

Elements ordering edited successfully.


Edit form

Elements
Handlers
Properties

Add elements to your form

Fields marked with * are required
Choose form element to add

(Content video)



Edit
Delete

Name (Text)*

Edit
Delete


Description (Textarea)

Edit
Delete

Website (URL)

Edit
Delete

(Content image)



Edit
Delete

Image (File)

No file selected.

Edit
Delete

Agree? (Boolean)

☐

Sed enim justo, blandit sodales nunc vitae, sollicitudin aliquet est. Aliquam tempor mattis efficitur? Phasellus accumsan, metus at porta varius, magna libero dictum massa, sit amet ornare leo augue et enim. Aliquam lobortis sit amet urna id cursus. Sed eu interdum nibh, ut dignissim lorem. Curabitur convallis augue lacus, a commodo lacus tincidunt vitae. Cras accumsan; lacus ut mollis luctus sed.

Edit
Delete

Save ordering

Build your forms Edit View

View form

Fields marked with * are required



Name*

Description

Website



Image

No file selected.

Agree?

☐

Sed enim justo, blandit sodales nunc vitae, sollicitudin aliquet est. Aliquam tempor mattis efficitur? Phasellus accumsan, metus at porta varius, magna libero dictum massa, sit amet ornare leo augue et enim. Aliquam lobortis sit amet urna id cursus. Sed eu interdum nibh, ut dignissim lorem. Curabitur convallis augue lacus, a commodo lacus tincidunt vitae. Cras accumsan; lacus ut mollis luctus sed.

© django-fobi example site 2014

38.2 Simple theme

38.2.1 View/edit form

Django administrationWelcome, test_admin. Log out

Home > Fobi > My test form


Edit form

ElementsHandlersProperties

Change form elementsAdd form element +

(Content video):

Delusional Insanity - To Far Beyond



Edit Delete

Name (Text):

Edit Delete


Description (Textarea):

Edit Delete

Website (URL):

Edit Delete

(Content image):



Edit Delete

Image (File):

Browse...No file selected.

Edit Delete

Agree? (Boolean):

☐

Sed enim justo, blandit sodales nunc vitae, sollicitudin aliquet est. Aliquam tempor mattis efficitur? Phasellus accumsan, metus at porta varius, magna libero dictum massa, sit amet ornare leo augue et enim. Aliquam lobortis sit amet urna id cursus. Sed eu interdum nibh, ut dignissim lorem. Curabitur convallis augue tunc, a conimodo tunc tristique vitae. Cras accumsan, tunc ut mollis tunc sed.

118Edit Delete

Save ordering

Django administration
Welcome, test_admin. Log out


Home » Fobi » My test form

Edit form

Elements
Handlers
Properties

Change form elements

Add form element +

(Content video):


Edit Delete

Name (Text):


Edit Delete

Description (Textarea):

Edit Delete

Website (URL):

Edit Delete

(Content image):


Edit Delete

Image (File):
 No file selected.

Edit Delete

Agree? (Boolean):
☐

Sed enim justo, blandit sodales nunc vitae, sollicitudin aliquet est. Aliquam tempor mattis efficitur? Phasellus accumsan, metus at porta varius, magna libero dictum massa, sit amet omare leo augue et enim. Aliquam lobortis sit amet urna id cursus. Sed eu interdum nibh, ut dignissim lorem. Curabitur convallis augue lacus, a commodo lacus tincidunt vitae. Cras accumsan; lacus ut mollis luctus sed.

Edit Delete

Save ordering

Django administration
Welcome, **test_admin**. Log out

Home > Fobi > My test form > Add "Hidden" element to the form

Add "Hidden" element to the form

Label:

Name:

Initial:

Max length:

Required: ☐

Add

Django administration
Welcome, **test_admin**. Log out

Home > Fobi > My test form

Edit form

Elements
Handlers
Properties

Change form handlers

Add form handler +

Handler	Actions
DB store	Delete
Mail ?	Edit Delete

Edit form

Elements **Handlers** Properties

Change form properties

Name:

Public?



Makes your form visible to the public.

Success page
title:

Custom message title to display after valid form is submitted

Success page
body:

Vivamus pharetra dui in tincidunt
dapibus! Mauris id congue tellus!
Vestibulum massa felis, varius ut
sem eu, sollicitudin lacinia enim.
Nam quis turpis ut purus mattis
mattis. Praesent ex orci, ultricies
ac blandit quis, pellentesque ac
magna. Aenean maximus sapien
tortor! Etiam vel consectetur orci;
in euismod elit! Cras ornare
sollicitudin blandit. Sed at turpis

Custom message text to display after valid form is submitted

Save

View form



Name:

Description:

Website:



Image: No file selected.

Agree? ☐

Sed enim justo, blandit sodales nunc vitae, sollicitudin aliquet est. Aliquam tempor mattis efficitur? Phasellus accumsan, metus at porta varius, magna libero dictum massa, sit amet ornare leo augue et enim. Aliquam lobortis sit amet urna id cursus. Sed eu interdum nibh, ut dignissim lorem. Curabitur convallis augue lacus, a commodo lacus tincidunt vitae. Cras accumsan; lacus ut mollis luctus sed.

(SUB)MODULES

Some additional documentation on `django-fobi` sub-modules are listed below.

39.1 `fobi.contrib.apps.djangocms_integration`

A `django-fobi` integration with DjangoCMS.

39.1.1 Prerequisites

Tested with DjangoCMS 2.4.3, 3.0.6 and 3.4.3 only. Might work on earlier (or later) versions as well.

39.1.2 Installation

39.1.2.1 Versions

See the requirements files:

- DjangoCMS 2.4.3 requirements.
- DjangoCMS 3.0.6 requirements.
- DjangoCMS 3.4.3 requirements.

39.1.2.2 `your_project/settings.py`

See the example settings files:

- DjangoCMS 2.4.3 settings.
- DjangoCMS 3.0.6/3.4.3. settings.

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    'cms',      # DjangoCMS

    'fobi.contrib.apps.djangocms_integration',  # Fobi DjangoCMS app
]
```

39.1.3 Information for developers

39.1.3.1 Templates for DjangoCMS

Django-CMS templates are quite specific and in some aspects are not 100% compatible with built-in themes (due to the fact that DjangoCMS intensively makes use of `django-sekizai` which isn't used in the generic templates).

That does not anyhow affect the core `fobi` and the built-in themes, although you can't magically reuse built-in `fobi` themes with Django-CMS (as it's done for [FeinCMS page templates](#)).

You would have to make custom page templates from scratch (or based on a copy of the desired bundled template). See how I did it for all bundled themes [here](#):

- [Bootstrap 3](#)
- [Foundation 5](#)
- [Simple](#)

39.1.3.2 Template rendering

The embed DjangoCMS widget is rendered with use of two theme templates:

- `view_embed_form_entry_ajax_template`: Used for rendering the form.
- `embed_form_entry_submitted_ajax_template`: Used for rendering the form sent event.

39.1.3.3 Using custom templates for rendering the widget

In the widget, you can specify a template which you want to be used for rendering the form or the form-sent event.

Example:

```
FOBI_DJANGOCMS_INTEGRATION_FORM_TEMPLATE_CHOICES = (  
    ("yourapp/custom_view_embed_form_v1.html",  
     "Custom embed form view template #1"),  
    ("yourapp/custom_view_embed_form_v2.html",  
     "Custom embed form view template #2"),  
)
```

Same goes for form-sent templates.

```
FOBI_DJANGOCMS_INTEGRATION_SUCCESS_PAGE_TEMPLATE_CHOICES = (  
    ("yourapp/custom_embed_form_submitted_v1.html",  
     "Custom form-sent template #1"),  
    ("yourapp/custom_embed_form_submitted_v2.html",  
     "Custom form-sent template #2"),  
)
```

Registering a template in the `FORM_TEMPLATE_CHOICES` makes it available for all the themes. If you rather want to use different custom templates for different themes, use the `FOBI_CUSTOM_THEME_DATA` as shown in the example below.

```
FOBI_CUSTOM_THEME_DATA = {  
    'bootstrap3': {  
        'djangoCMS_integration': {  
            'form_template_choices': [  

```

(continues on next page)

(continued from previous page)

```

        ('fobi/bootstrap3_extras/view_embed_form.html',
         "Custom bootstrap3 embed form view template"),
    ],
    'success_page_template_choices': [
        ('fobi/bootstrap3_extras/embed_form_submitted.html',
         "Custom bootstrap3 embed form entry submitted template"),
    ],
},
},
'foundation5': {
    'djangocms_integration': {
        'form_template_choices': [
            ('fobi/foundation5_extras/view_embed_form.html',
             "Custom foundation5 embed form view template"),
        ],
        'success_page_template_choices': [
            ('fobi/foundation5_extras/embed_form_submitted.html',
             "Custom foundation5 embed form entry submitted template"),
        ],
    },
},
},
}

```

39.1.4 Usage

The `fobi.contrib.apps.djangocms_integration.models.FobiFormWidget` consists of the following fields:

- **Form:** The form to be used.
- **Form template name:** Template to be used to render the embed form.
- **Hide form title:** If checked, no form title would be shown.
- **Form title:** Overrides the standard form title.
- **Submit button text:** Overrides the default submit button text.
- **Success page template name:** Template to be used to render the embed form-sent event.
- **Hide success page title:** If checked, no form-sent title would be shown.
- **Success page title:** Overrides the form-sent title.
- **Success page text:** Overrides the form-sent text.

39.2 fobi.contrib.apps.drf_integration

A django-fobi integration with Django REST framework.

Supported actions are:

- *LIST*: List all the forms.
- *OPTIONS*: Describe the given form.
- *PUT*: Submit form data.

39.2.1 Live demo

Live demo is available on Heroku.

- [The core](#)
- [Django REST framework integration](#)

39.2.2 Supported fields

The following fields are supported.

39.2.2.1 Content (presentational form elements)

Unlike standard fields, content fields are purely presentational. You're not supposed to make write actions on them (it won't work). Neither will they be displayed in the browsable API (list/retrieve actions). However, they will be listed in the options action call. All content fields are of type "content".

- `content_image`
- `content_image_url`
- `content_richtext`
- `content_text`
- `content_video`

39.2.2.2 Fields

- `boolean`
- `checkbox_select_multiple`
- `date`
- `date_drop_down`
- `datetime`
- `decimal`
- `duration`
- `email`
- `file`
- `float`
- `hidden` (in terms of the Django REST framework - a read-only field)
- `input` (some sort of a copy of `text` plugin)
- `integer`
- `ip_address`
- `null_boolean`
- `password` (some sort of a copy of `text` plugin)
- `radio`

- `range_select`
- `regex`
- `select`
- `select_multiple`
- `select_multiple_with_max`
- `slider` (just a copy of `range_select`, for compatibility with main package)
- `slug`
- `text`
- `textarea` (some sort of a copy of `text` plugin)
- `time`
- `url`

39.2.3 Not (yet) supported fields

The following fields are not supported. Those marked with asterisk are planned to be supported in the upcoming releases.

- `select_model_object`
- `select_mptt_model_object`
- `select_multiple_model_objects`
- `select_multiple_mptt_model_objects`

39.2.4 Implementation details

Each `django-fobi` plugin has its' own representative integration plugin within `fobi.contrib.apps.drf_integration` package.

Some of the plugins may seem to have zero-added-value and in fact they are. For instance, DRF integration `slider` plugin is just an exact copy of the `range_select` plugin, created in order to provide exactly the same form fields generated in the API.

You should mention all the plugins you want to use explicitly in the project settings. Thus, if you have used (included in the `INSTALLED_APPS`) the core plugins:

- `fobi.contrib.plugins.form_elements.fields.boolean`
- `fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple`
- `fobi.contrib.plugins.form_elements.fields.date`
- `fobi.contrib.plugins.form_elements.fields.date_drop_down`
- `fobi.contrib.plugins.form_elements.fields.datetime`
- `fobi.contrib.plugins.form_elements.fields.decimal`
- `fobi.contrib.plugins.form_elements.fields.duration`
- `fobi.contrib.plugins.form_elements.fields.email`
- `fobi.contrib.plugins.form_elements.fields.file`

- `fobi.contrib.plugins.form_elements.fields.float`
- `fobi.contrib.plugins.form_elements.fields.hidden`
- `fobi.contrib.plugins.form_elements.fields.input`
- `fobi.contrib.plugins.form_elements.fields.integer`
- `fobi.contrib.plugins.form_elements.fields.ip_address`
- `fobi.contrib.plugins.form_elements.fields.null_boolean`
- `fobi.contrib.plugins.form_elements.fields.password`
- `fobi.contrib.plugins.form_elements.fields.radio`
- `fobi.contrib.plugins.form_elements.fields.range_select`
- `fobi.contrib.plugins.form_elements.fields.regex`
- `fobi.contrib.plugins.form_elements.fields.select`
- `fobi.contrib.plugins.form_elements.fields.select_multiple`
- `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max`
- `fobi.contrib.plugins.form_elements.fields.slider`
- `fobi.contrib.plugins.form_elements.fields.slug`
- `fobi.contrib.plugins.form_elements.fields.text`
- `fobi.contrib.plugins.form_elements.fields.textarea`
- `fobi.contrib.plugins.form_elements.fields.time`
- `fobi.contrib.plugins.form_elements.fields.url`
- `fobi.contrib.plugins.form_elements.content.content_image`
- `fobi.contrib.plugins.form_elements.content.content_image_url`
- `fobi.contrib.plugins.form_elements.content.content_richtext`
- `fobi.contrib.plugins.form_elements.content.content_text`
- `fobi.contrib.plugins.form_elements.content.content_video`
- `fobi.contrib.plugins.form_handlers.db_store`
- `fobi.contrib.plugins.form_handlers.http_repost`
- `fobi.contrib.plugins.form_handlers.mail`

You should include their correspondent Django REST framework implementations in the `INSTALLED_APPS` as well:

- `fobi.contrib.apps.drf_integration.form_elements.fields.boolean`
- `fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple`
- `fobi.contrib.apps.drf_integration.form_elements.fields.date`
- `fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down`
- `fobi.contrib.apps.drf_integration.form_elements.fields.datetime`
- `fobi.contrib.apps.drf_integration.form_elements.fields.decimal`
- `fobi.contrib.apps.drf_integration.form_elements.fields.duration`
- `fobi.contrib.apps.drf_integration.form_elements.fields.email`

- `fobi.contrib.apps.drf_integration.form_elements.fields.file`
- `fobi.contrib.apps.drf_integration.form_elements.fields.float`
- `fobi.contrib.apps.drf_integration.form_elements.fields.hidden`
- `fobi.contrib.apps.drf_integration.form_elements.fields.input`
- `fobi.contrib.apps.drf_integration.form_elements.fields.integer`
- `fobi.contrib.apps.drf_integration.form_elements.fields.ip_address`
- `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean`
- `fobi.contrib.apps.drf_integration.form_elements.fields.password`
- `fobi.contrib.apps.drf_integration.form_elements.fields.radio`
- `fobi.contrib.apps.drf_integration.form_elements.fields.range_select`
- `fobi.contrib.apps.drf_integration.form_elements.fields.regex`
- `fobi.contrib.apps.drf_integration.form_elements.fields.select`
- `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple`
- `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max`
- `fobi.contrib.apps.drf_integration.form_elements.fields.slider`
- `fobi.contrib.apps.drf_integration.form_elements.fields.slug`
- `fobi.contrib.apps.drf_integration.form_elements.fields.text`
- `fobi.contrib.apps.drf_integration.form_elements.fields.textarea`
- `fobi.contrib.apps.drf_integration.form_elements.fields.time`
- `fobi.contrib.apps.drf_integration.form_elements.fields.url`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_image`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_image_url`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_richtext`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_text`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_video`
- `fobi.contrib.apps.drf_integration.form_handlers.db_store`
- `fobi.contrib.apps.drf_integration.form_handlers.http_repost`
- `fobi.contrib.apps.drf_integration.form_handlers.mail`

39.2.5 Installation

39.2.5.1 Versions

Was made with `django-rest-framework` 3.6.2. May work on earlier versions, although not guaranteed.

See the [requirements file](#).

39.2.5.2 your_project/settings.py

See the example settings file.

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    # ...
    # Here should come a list of form element plugins of the core
    # package, followed by the list of form handler plugins of the core
    # package, followed by the list of themes of the core package and
    # all other apps that do matter.
    # ...
    'rest_framework', # Django REST framework
    'fobi.contrib.apps.drf_integration', # DRF integration app

    # DRF integration form element plugins - fields
    'fobi.contrib.apps.drf_integration.form_elements.fields.boolean',
    'fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple',
    'fobi.contrib.apps.drf_integration.form_elements.fields.date',
    'fobi.contrib.apps.drf_integration.form_elements.fields.datetime',
    'fobi.contrib.apps.drf_integration.form_elements.fields.decimal',
    'fobi.contrib.apps.drf_integration.form_elements.fields.duration',
    'fobi.contrib.apps.drf_integration.form_elements.fields.email',
    'fobi.contrib.apps.drf_integration.form_elements.fields.file',
    'fobi.contrib.apps.drf_integration.form_elements.fields.float',
    'fobi.contrib.apps.drf_integration.form_elements.fields.hidden',
    'fobi.contrib.apps.drf_integration.form_elements.fields.input',
    'fobi.contrib.apps.drf_integration.form_elements.fields.integer',
    'fobi.contrib.apps.drf_integration.form_elements.fields.ip_address',
    'fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean',
    'fobi.contrib.apps.drf_integration.form_elements.fields.password',
    'fobi.contrib.apps.drf_integration.form_elements.fields.radio',
    'fobi.contrib.apps.drf_integration.form_elements.fields.range_select',
    'fobi.contrib.apps.drf_integration.form_elements.fields.regex',
    'fobi.contrib.apps.drf_integration.form_elements.fields.select',
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple',
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max',
    'fobi.contrib.apps.drf_integration.form_elements.fields.slider',
    'fobi.contrib.apps.drf_integration.form_elements.fields.slug',
    'fobi.contrib.apps.drf_integration.form_elements.fields.text',
    'fobi.contrib.apps.drf_integration.form_elements.fields.textarea',
    'fobi.contrib.apps.drf_integration.form_elements.fields.time',
    'fobi.contrib.apps.drf_integration.form_elements.fields.url',

    # DRF integration form element plugins - presentational
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image',
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image_url',
    'fobi.contrib.apps.drf_integration.form_elements.content.content_richtext',
    'fobi.contrib.apps.drf_integration.form_elements.content.content_text',
    'fobi.contrib.apps.drf_integration.form_elements.content.content_video',

    # DRF integration form handler plugins
    'fobi.contrib.apps.drf_integration.form_handlers.db_store',
```

(continues on next page)

(continued from previous page)

```
'fobi.contrib.apps.drf_integration.form_handlers.mail',
'fobi.contrib.apps.drf_integration.form_handlers.http_repost',
# ...
]
```

39.2.5.3 your_project/urls.py

Add the following code to the main `urls.py` of your project:

```
# Conditionally including django-rest-framework integration app
if 'fobi.contrib.apps.drf_integration' in settings.INSTALLED_APPS:
    from fobi.contrib.apps.drf_integration.urls import fobi_router
    urlpatterns += [
        url(r'^api/', include(fobi_router.urls))
    ]
```

39.2.6 Usage

If you have followed the steps above precisely, you would be able to access the API using `http://localhost:8000/api/fobi-form-entry/`.

Actions/methods supported:

39.2.6.1 LIST

```
GET /api/fobi-form-entry/
```

Lists all the forms available. Anonymous users would see the list of all public forms. Authenticated users would see their own forms in addition to the public forms.

39.2.6.2 OPTIONS

```
OPTIONS /api/fobi-form-entry/{FORM_SLUG}/
```

Lists all field options for the selected form.

See the [test DRF form](#) and [same form in DRF integration app](#) with most of the fields that do have rich additional metadata.

OPTIONS call produces the following response:

```
OPTIONS /api/fobi-form-entry/test-drf-form/
HTTP 200 OK
Allow: GET, PUT, PATCH, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
{
  "name": "Fobi Form Entry Instance",
  "description": "FormEntry view set.",
  "renders": [
    "application/json",
    "text/html"
  ],
  "parses": [
    "application/json",
    "application/x-www-form-urlencoded",
    "multipart/form-data"
  ],
  "actions": {
    "PUT": {
      "test_integer": {
        "type": "integer",
        "required": false,
        "read_only": false,
        "label": "Test integer",
        "min_value": 1,
        "max_value": 20,
        "initial": 10
      },
      "test_email": {
        "type": "email",
        "required": true,
        "read_only": false,
        "label": "Test email",
        "help_text": "Donec mollis hendrerit risus. Phasellus a "
          "est. Nam ipsum risus, rutrum vitae, "
          "vestibulum eu, molestie vel, lacus. "
          "Praesent nec nisl a purus blandit viverra. "
          "Cras id dui.",
        "max_length": 255,
        "placeholder": "john@doe.com"
      },
      "test_text": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Test text",
        "help_text": "Sed lectus. Phasellus gravida semper "
          "nisi. Curabitur at lacus ac velit ornare "
          "lobortis. Mauris turpis nunc, blandit et, "
          "volutpat molestie, porta ut, ligula. Lorem "
          "ipsum dolor sit amet, consectetur "
          "adipiscing elit.",
        "max_length": 255,
        "placeholder": "Lorem ipsum dolor sit amet"
      },
      "test_url": {
        "type": "url",
        "required": false,
```

(continues on next page)

(continued from previous page)

```

        "read_only": false,
        "label": "Test URL",
        "max_length": 255,
        "initial": "http://github.com"
    },
    "test_decimal_field": {
        "type": "decimal",
        "required": false,
        "read_only": false,
        "label": "Test decimal field",
        "min_value": 1.0,
        "max_value": 25.0,
        "initial": 10.0,
        "placeholder": "3.14",
        "max_digits": 5,
        "decimal_places": 2
    },
    "test_float_field": {
        "type": "float",
        "required": false,
        "read_only": false,
        "label": "Test float field",
        "min_value": 1.0,
        "max_value": 10.0,
        "initial": 3.14
    },
    "test_ip_address": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Test IP address",
        "max_length": 255,
        "placeholder": "127.0.0.1"
    },
    "test_password_field": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Test password field",
        "max_length": 255,
        "placeholder": "your-secret-password"
    },
    "test_regex_field": {
        "type": "regex",
        "required": false,
        "read_only": false,
        "label": "Test regex field",
        "max_length": 255,
        "regex": "^[a-zA-Z]+$"
    },
    "test_slug_field": {
        "type": "slug",

```

(continues on next page)

(continued from previous page)

```

        "required": false,
        "read_only": false,
        "label": "Test slug field",
        "max_length": 255,
        "placeholder": "lorem-ipsum-dolor-sit-amet"
    },
    "test_textarea_field": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Test textarea field",
        "placeholder": "Pellentesque habitant morbi tristique."
    },
    "test_input_field": {
        "type": "string",
        "required": false,
        "read_only": true,
        "label": "Test input field",
        "max_length": 255,
        "autofocus": "autofocus",
        "autocomplete": "on",
        "disabled": "disabled"
    },
    "content_image_url_b0996b16-9f1c-430d-a6c7-0a722f4c2177": {
        "type": "content",
        "required": false,
        "read_only": true,
        "initial": "<p><img src=\"http://example.com/image.jpg\" alt=\"n.n.\" ↵
↵width=\"600\"/></p>",
        "contenttype": "image",
        "raw_data": {
            "url": "http://example.com/image.jpg",
            "alt": "n.n.",
            "fit_method": "fit_width",
            "size": "600x600"
        },
        "content": "<p><img src=\"http://example.com/image.jpg\" alt=\"n.n.\" ↵
↵width=\"600\"/></p>"
    },
    "content_text_de4d69b2-99e1-479d-8c61-1534dea7c981": {
        "type": "content",
        "required": false,
        "read_only": true,
        "initial": "<p>Pellentesque posuere. Quisque id mi. "
            "Duis arcu tortor, suscipit eget, imperdiet "
            "nec, imperdiet iaculis, ipsum. Phasellus a "
            "est. In turpis.</p>",
        "contenttype": "text",
        "raw_data": {
            "text": "Pellentesque posuere. Quisque id mi. Duis "
                "arcu tortor, suscipit eget, imperdiet nec, "
                "imperdiet iaculis, ipsum. Phasellus a est. "
    
```

(continues on next page)

(continued from previous page)

```

        "In turpis."
    },
    "content": "<p>Pellentesque posuere. Quisque id mi. Duis "
               "arcu tortor, suscipit eget, imperdiet nec, "
               "imperdiet iaculis, ipsum. Phasellus a est. "
               "In turpis.</p>"
    },
    "content_video_f4799aca-9a0b-4f1a-8069-dda611858ef4": {
        "type": "content",
        "required": false,
        "read_only": true,
        "initial": "<iframe src=\"//www.youtube.com/embed/8GVIui0JK0M\" width=\
↪\"500\" height=\"400\" frameborder=\"0\" allowfullscreen></iframe>",
        "contenttype": "video",
        "raw_data": {
            "title": "Delusional Insanity - To far beyond...",
            "url": "https://www.youtube.com/watch?v=8GVIui0JK0M&t=1s",
            "size": "500x400"
        },
        "content": "<iframe src=\"//www.youtube.com/embed/8GVIui0JK0M\" width=\
↪\"500\" height=\"400\" frameborder=\"0\" allowfullscreen></iframe>"
    }
}

```

Some insights:

Meta-data is passed to the `DRFIntegrationFormElementPluginProcessor` as `field_metadata` argument, which is supposed to be a dict.

- Example 1: content_image plugin
- Example 2: decimal plugin
- Example 3: text plugin

Private forms would be only visible to authenticated users.

39.2.6.3 PUT

```
PUT /api/fobi-form-entry/{FORM_SLUG}/  
  
{DATA}
```

39.2.7 Callbacks

Callbacks work just the same way the core callbacks work.

39.2.7.1 fobi_form_callbacks.py

```
from fobi.base import (  
    integration_form_callback_registry,  
    IntegrationFormCallback,  
)  
  
from fobi.constants import (  
    CALLBACK_BEFORE_FORM_VALIDATION,  
    CALLBACK_FORM_INVALID,  
    CALLBACK_FORM_VALID,  
    CALLBACK_FORM_VALID_AFTER_FORM_HANDLERS,  
    CALLBACK_FORM_VALID_BEFORE_SUBMIT_PLUGIN_FORM_DATA,  
)  
  
from fobi.contrib.apps.drf_integration import UID as INTEGRATE_WITH  
  
class DRFSaveAsFooItem(IntegrationFormCallback):  
    """Save the form as a foo item, if certain conditions are met."""  
  
    stage = CALLBACK_FORM_VALID  
    integrate_with = INTEGRATE_WITH  
  
    def callback(self, form_entry, request, **kwargs):  
        """Custom callback login comes here."""  
        logger.debug("Great! Your form is valid!")  
  
class DRFDummyInvalidCallback(IntegrationFormCallback):  
    """Saves the form as a foo item, if certain conditions are met."""  
  
    stage = CALLBACK_FORM_INVALID  
    integrate_with = INTEGRATE_WITH  
  
    def callback(self, form_entry, request, **kwargs):  
        """Custom callback login comes here."""  
        logger.debug("Damn! You've made a mistake, boy!")
```


39.2.8 Testing

To test Django REST framework integration package only, run the following command:

```
./runtests.py src/fobi/tests/test_drf_integration.py
```

or use plain Django tests:

```
./manage.py test fobi.tests.test_drf_integration --settings=settings.test
```

39.2.9 Limitations

Certain fields are not available yet (relational fields).

39.3 Form elements

39.3.1 Content form element

Presentational form elements for `drf_integration`.

39.3.1.1 fobi.contrib.apps.drf_integration.form_elements.content.content_image

A django-fobi ContentImage plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentImage`.

39.3.1.1.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_image` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.1.1.2 Usage

Unlike standard fields, `ContentImage` field is purely presentational. You're not supposed to make write actions on it (it won't work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
"actions": {
  "PUT": {
    // ...
    "content_image_89c8c319-195b-487a-a44d-f59ef14a5d44": {
      "type": "content",
      "required": false,
      "read_only": true,
      "contenttype": "image",
      "content": "\n<p>\n\n\n\n<img src=\"/media/fobi_plugins/content_image/test-
↪image-thumbnail.jpg\" alt=\"Lorem ipsum\"/>\n\n\n</p>\n",
      "raw": {
        "file": "/media/fobi_plugins/content_image/test-image.jpg",
        "alt": "Lorem ipsum",
        "fit_method": "center",
        "size": "500x500"
      }
    },
    // ...
  }
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “image” for `ContentImage` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentImage` plugin. Contains “file”, “alt”, “fit_method” and “size” attributes.

39.3.1.2 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url

A django-fobi `ContentImageURL` plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentImage`.

39.3.1.2.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_image_url` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image_url',
```

(continues on next page)

(continued from previous page)

```
# ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.1.2.2 Usage

Unlike standard fields, `ContentImageURL` field is purely presentational. You’re not supposed to make write actions on it (it won’t work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
"actions": {
  "PUT": {
    // ...
    "content_image_89c8c319-195b-487a-a44d-f59ef14a5d44": {
      "type": "content",
      "required": false,
      "read_only": true,
      "contenttype": "image",
      "content": "\n<p>\n\n\n\n<img src=\"http://example.com/media/test-image.
→jpg\" alt=\"Lorem ipsum\"/>\n\n\n</p>\n",
      "raw": {
        "url": "http://example.com/media/test-image.jpg",
        "alt": "Lorem ipsum",
        "fit_method": "fit_width",
        "size": "500x500"
      }
    },
    // ...
  }
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “image” for `ContentImageURL` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentImageURL` plugin. Contains “url”, “alt”, “fit_method” and “size” attributes.

39.3.1.3 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown

A django-fobi ContentMarkdown plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentMarkdown`.

39.3.1.3.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_markdown` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.content.content_markdown',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.1.3.2 Usage

Unlike standard fields, `ContentMarkdown` field is purely presentational. You're not supposed to make write actions on it (it won't work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
{  
  "actions": {  
    "PUT": {  
      // ...  
      "content_markdown_89c8c319-195b-487a-a44d-f59ef14a5d44": {  
        "type": "content",  
        "required": false,  
        "read_only": true,  
        "contenttype": "text",  
        "content": "\n<p>\n\nLorem ipsum dolor sit amet.\n\n</p>\n",  
        "raw": {  
          "text": "Lorem ipsum dolor sit amet."  
        }  
      },  
      // ...  
    }  
  }  
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “text” for `ContentText` field.
- `content` (str): Representation of the content. Rendered partial HTML.

- `raw` (json dict): Raw attributes of the `ContentText` plugin. Contains “text” attribute.

39.3.1.4 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext

A `django-fobi` `ContentRichText` plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentRichText`.

39.3.1.4.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_richtext` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.content.content_richtext',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.1.4.2 Usage

Unlike standard fields, `ContentText` field is purely presentational. You’re not supposed to make write actions on it (it won’t work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
{
  "actions": {
    "PUT": {
      // ...
      "content_text_89c8c319-195b-487a-a44d-f59ef14a5d44": {
        "type": "content",
        "required": false,
        "read_only": true,
        "contenttype": "text",
        "content": "\n<p>\n\nLorem ipsum dolor sit amet.\n\n\n</p>\n",
        "raw": {
          "text": "Lorem ipsum dolor sit amet."
        }
      },
      // ...
    }
  }
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.

- `contenttype` (str): Set to “text” for `ContentText` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentText` plugin. Contains “text” attribute.

39.3.1.5 fobi.contrib.apps.drf_integration.form_elements.content.content_text

A django-fobi `ContentText` plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentText`.

39.3.1.5.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.content.content_text',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.1.5.2 Usage

Unlike standard fields, `ContentText` field is purely presentational. You’re not supposed to make write actions on it (it won’t work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
{  
  "actions": {  
    "PUT": {  
      // ...  
      "content_text_89c8c319-195b-487a-a44d-f59ef14a5d44": {  
        "type": "content",  
        "required": false,  
        "read_only": true,  
        "contenttype": "text",  
        "content": "\n<p>\n\nLorem ipsum dolor sit amet.\n\n</p>\n",  
        "raw": {  
          "text": "Lorem ipsum dolor sit amet."  
        }  
      },  
    },  
    // ...  
  }  
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “text” for `ContentText` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentText` plugin. Contains “text” attribute.

39.3.1.6 fobi.contrib.apps.drf_integration.form_elements.content.content_video

A django-fobi `ContentVideo` plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentVideo`.

39.3.1.6.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_video` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.content.content_video',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.1.6.2 Usage

Unlike standard fields, `ContentVideo` field is purely presentational. You’re not supposed to make write actions on it (it won’t work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
"actions": {
  "PUT": {
    // ...
    "content_video_41a6b951-e6f9-4f08-ada6-3b109aa9a72f": {
      "type": "content",
      "required": false,
      "read_only": true,
      "contenttype": "video",
      "content": "\n<iframe src=\"//www.youtube.com/embed/3P1qcVcs4Ik\" width=\n
→ "500\" height=\"400\" frameborder=\"0\" allowfullscreen></iframe>\n",
      "raw": {
        "title": "Cras risus ipsum faucibus",
        "url": "https://www.youtube.com/watch?v=3P1qcVcs4Ik",
```

(continues on next page)

(continued from previous page)

```
        "size": "500x400"
    },
    // ...
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “video” for `ContentVideo` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentVideo` plugin. Contains “title”, “url” and “size” attributes.

39.3.2 Fields

Form fields for `drf_integration`.

39.3.2.1 fobi.contrib.apps.drf_integration.form_elements.fields.boolean

A django-fobi BooleanField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.BooleanField`.

39.3.2.1.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.boolean',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.2 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.MultipleChoiceField`.

39.3.2.2.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.3 fobi.contrib.apps.drf_integration.form_elements.fields.date

A django-fobi DateField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DateField`.

39.3.2.3.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.date` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.date',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.4 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down

A django-fobi DateField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DateField`.

This plugin has been made primarily for compatibility with `date_drop_down` plugin of the core package.

39.3.2.4.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.5 fobi.contrib.apps.drf_integration.form_elements.fields.datetime

A django-fobi DateField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DateTimeField`.

39.3.2.5.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.datetime` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.datetime',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.6 fobi.contrib.apps.drf_integration.form_elements.fields.decimal

A django-fobi DecimalField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DecimalField`.

39.3.2.6.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.decimal` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.decimal',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.7 fobi.contrib.apps.drf_integration.form_elements.fields.duration

A django-fobi DurationField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DurationField`.

39.3.2.7.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.duration` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.duration',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.8 fobi.contrib.apps.drf_integration.form_elements.fields.email

A django-fobi EmailField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.EmailField`.

39.3.2.8.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.email` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.email',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.9 fobi.contrib.apps.drf_integration.form_elements.fields.file

A django-fobi FileField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.FileField`.

39.3.2.9.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.file` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.file',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.10 fobi.contrib.apps.drf_integration.form_elements.fields.float

A django-fobi FloatField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.FloatField`.

39.3.2.10.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.float` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.float',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.11 fobi.contrib.apps.drf_integration.form_elements.fields.hidden

A django-fobi HiddenField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.HiddenField`.

Note, that in terms of the Django REST framework it is a read-only field. Any values posted along won't be saved. Initial value would.

39.3.2.11.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.hidden` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.hidden',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.12 `fobi.contrib.apps.drf_integration.form_elements.fields.input`

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

39.3.2.12.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.input` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.input',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.13 `fobi.contrib.apps.drf_integration.form_elements.fields.integer`

A django-fobi IntegerField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.IntegerField`.

39.3.2.13.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.integer` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.integer',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.14 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address

A django-fobi IPAddressField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.IPAddressField`.

39.3.2.14.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.ip_address` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.ip_address',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.15 fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean

A django-fobi NullBooleanField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.NullBooleanField`.

39.3.2.15.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.16 fobi.contrib.apps.drf_integration.form_elements.fields.password

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

39.3.2.16.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.password` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.password',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.17 fobi.contrib.apps.drf_integration.form_elements.fields.radio

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

39.3.2.17.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.radio` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.radio',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.18 fobi.contrib.apps.drf_integration.form_elements.fields.range_select

A django-fobi ChoiceField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

39.3.2.18.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.range_select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.range_select',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.19 fobi.contrib.apps.drf_integration.form_elements.fields.regex

A django-fobi RegexField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.RegexField`.

39.3.2.19.1 Installation

1. Add `fobi.contrib.apps.drf_integration.form_elements.fields.regex` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.regex',  
    # ...  
)
```

2. In the terminal type:

```
./manage.py fobi_sync_plugins
```

3. Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.20 fobi.contrib.apps.drf_integration.form_elements.fields.select

A django-fobi ChoiceField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

39.3.2.20.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.select',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.21 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object

A django-fobi ModelChoiceField plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.py.ModelChoiceField`.

39.3.2.21.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.22 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.MultipleChoiceField`.

39.3.2.22.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.23 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects

A django-fobi MultipleModelChoiceField plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.py.MultipleModelChoiceField`.

39.3.2.23.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_
↪objects',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.24 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.MultipleChoiceField`.

39.3.2.24.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.25 fobi.contrib.apps.drf_integration.form_elements.fields.slider

A django-fobi ChoiceField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

39.3.2.25.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.slider` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.slider',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.26 fobi.contrib.apps.drf_integration.form_elements.fields.slug

A django-fobi SlugField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.SlugField`.

39.3.2.26.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.slug` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.slug',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.27 fobi.contrib.apps.drf_integration.form_elements.fields.text

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

39.3.2.27.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.text',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.28 fobi.contrib.apps.drf_integration.form_elements.fields.textarea

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

39.3.2.28.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.textarea` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.textarea',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.29 fobi.contrib.apps.drf_integration.form_elements.fields.time

A django-fobi TimeField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.TimeField`.

39.3.2.29.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.time` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.time',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.3.2.30 fobi.contrib.apps.drf_integration.form_elements.fields.url

A django-fobi URLField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.URLField`.

39.3.2.30.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.time` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.url',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.4 Form handlers

Form handlers for `drf_integration`.

39.4.1 fobi.contrib.apps.drf_integration.form_handlers.db_store

A django-fobi Mail form handler plugin for integration with Django REST framework. Saves submitted form data into the `SavedFormDataRow` model.

39.4.1.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_handlers.db_store` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_handlers.db_store',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py migrate
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.4.2 fobi.contrib.apps.drf_integration.form_handlers.http_repost

A django-fobi HTTP repost form handler plugin for integration with Django REST framework. Submits the form to another endpoint specified.

39.4.2.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_handlers.http_repost` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_handlers.http_repost',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.4.3 fobi.contrib.apps.drf_integration.form_handlers.mail

A django-fobi Mail form handler plugin for integration with Django REST framework. Submits the form data by email to the specified email address.

39.4.3.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_handlers.mail` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_handlers.mail',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py migrate  
  
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.4.4 fobi.contrib.apps.drf_integration.form_handlers.mail_sender

A django-fobi Mail the sender form handler plugin for integration with Django REST framework. Submits the form data by email to the specified email address.

39.4.4.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_handlers.mail_sender` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_handlers.mail_sender',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py migrate  
  
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

39.5 fobi.contrib.apps.feincms_integration

A django-fobi integration with FeinCMS.

39.5.1 Prerequisites

Tested with FeinCMS 1.10, 1.12 and 1.13 only. Might work on earlier (or later) versions as well.

39.5.2 Installation

39.5.2.1 Versions

See the requirements files:

- [FeinCMS 1.10 requirements](#).
- [FeinCMS 1.12 requirements](#).
- [FeinCMS 1.13 requirements](#).

39.5.2.2 your_project/settings.py

See the example settings file.

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    'feincms', # FeinCMS

    'fobi.contrib.apps.feincms_integration', # Fobi FeinCMS app

    'page', # Example
]

FEINCMS_RICHTEXT_INIT_CONTEXT = {
    'TINYMCE_JS_URL': STATIC_URL + 'tiny_mce/tiny_mce.js',
}
```

39.5.2.3 your_project/page/models.py

```
from django.utils.translation import ugettext_lazy as _

from feincms.module.page.models import Page
from feincms.content.raw.models import RawContent
from feincms.content.richtext.models import RichTextContent
# Import the ``django-fobi`` widget.
from fobi.contrib.apps.feincms_integration.widgets import FobiFormWidget

Page.register_extensions('feincms.module.extensions.translations',)

# Register basic template.
Page.register_templates(
    {
        'title': _(u"Base template"),
        'path': 'page/base.html',
        'key': 'page_base',
        'regions': (
            ('main', _(u"Main")),
            ('sidebar', _(u"Sidebar")),
        )
    },
)

# Standard content types
Page.create_content_type(RawContent)
Page.create_content_type(RichTextContent)

# Register the ``django-fobi`` widget.
Page.create_content_type(FobiFormWidget)
```

39.5.2.4 your_project/admin.py

```
from django.contrib import admin

from feincms.module.page.modeladmins import PageAdmin

from page.models import Page

admin.site.register(Page, PageAdmin)
```

39.5.2.5 Information for developers

39.5.2.5.1 Template rendering

The embed FeinCMS widget is rendered with use of two theme templates:

- `view_embed_form_entry_ajax_template`: Used for rendering the form.
- `embed_form_entry_submitted_ajax_template`: Used for rendering the form sent event.

39.5.2.6 Using custom templates for rendering the widget

In the widget, you can specify a template which you want to be used for rendering the form or the form-sent event.

Example:

```
FOBI_FEINCMS_INTEGRATION_FORM_TEMPLATE_CHOICES = (
    ("yourapp/custom_view_embed_form_v1.html",
     "Custom embed form view template #1"),
    ("yourapp/custom_view_embed_form_v2.html",
     "Custom embed form view template #2"),
)
```

Same goes for form-sent templates.

```
FOBI_FEINCMS_INTEGRATION_SUCCESS_PAGE_TEMPLATE_CHOICES = (
    ("yourapp/custom_embed_form_submitted_v1.html",
     "Custom form-sent template #1"),
    ("yourapp/custom_embed_form_submitted_v2.html",
     "Custom form-sent template #2"),
)
```

Registering a template in the `FORM_TEMPLATE_CHOICES` makes it available for all the themes. If you rather want to use different custom templates for different themes, use the `FOBI_CUSTOM_THEME_DATA` as shown in the example below.

```
FOBI_CUSTOM_THEME_DATA = {
    'bootstrap3': {
        'feincms_integration': {
            'form_template_choices': [
                ('fobi/bootstrap3_extras/view_embed_form.html',
                 "Custom bootstrap3 embed form view template"),
            ],
            'success_page_template_choices': [
```

(continues on next page)

(continued from previous page)

```
        ('fobi/bootstrap3_extras/embed_form_submitted.html',
         "Custom bootstrap3 embed form entry submitted template"),
    ],
},
},
'foundation5': {
    'feincms_integration': {
        'form_template_choices': [
            ('fobi/foundation5_extras/view_embed_form.html',
             "Custom foundation5 embed form view template"),
        ],
        'success_page_template_choices': [
            ('fobi/foundation5_extras/embed_form_submitted.html',
             "Custom foundation5 embed form entry submitted template"),
        ],
    },
},
},
}
```

39.5.3 Usage

The `fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget` consists of the following fields:

- Form: The form to be used.
- Form template name: Template to be used to render the embed form.
- Hide form title: If checked, no form title would be shown.
- Form title: Overrides the standard form title.
- Submit button text: Overrides the default submit button text.
- Success page template name: Template to be used to render the embed form-sent event.
- Hide success page title: If checked, no form-sent title would be shown.
- Success page title: Overrides the form-sent title.
- Success page text: Overrides the form-sent text.

39.6 fobi.contrib.apps.mezzanine_integration

A django-fobi integration with Mezzanine.

39.6.1 Prerequisites

Tested with Mezzanine 3.1.10 and 4.2.3 only. Might work on earlier (or later) versions as well.

39.6.2 Installation

39.6.2.1 Versions

See the requirements files:

- [Mezzanine 4.2.3 requirements](#).

39.6.2.2 `your_project/settings.py`

See the [example settings file](#).

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    # Standard mezzanine apps

    'fobi.contrib.apps.mezzanine_integration', # Fobi Mezzanine app
]
```

39.6.3 Information for developers

39.6.3.1 Template rendering

The form embed into Mezzanine page is rendered with use of two theme templates:

- `view_embed_form_entry_ajax_template`: Used for rendering the form.
- `embed_form_entry_submitted_ajax_template`: Used for rendering the form sent event.

39.6.3.2 Using custom templates for rendering the form

In the widget, you can specify a template which you want to be used for rendering the form or the form-sent event.

Example:

```
FOBI_MEZZANINE_INTEGRATION_FORM_TEMPLATE_CHOICES = (
    ("yourapp/custom_view_embed_form_v1.html",
     "Custom embed form view template #1"),
    ("yourapp/custom_view_embed_form_v2.html",
     "Custom embed form view template #2"),
)
```

Same goes for form-sent templates.

```
FOBI_MEZZANINE_INTEGRATION_SUCCESS_PAGE_TEMPLATE_CHOICES = (
    ("yourapp/custom_embed_form_submitted_v1.html",
     "Custom form-sent template #1"),
    ("yourapp/custom_embed_form_submitted_v2.html",
```

(continues on next page)

(continued from previous page)

```
        "Custom form-sent template #2"),
    )
```

Registering a template in the `FORM_TEMPLATE_CHOICES` makes it available for all the themes. If you rather want to use different custom templates for different themes, use the `FOBI_CUSTOM_THEME_DATA` as shown in the example below.

```
FOBI_CUSTOM_THEME_DATA = {
    'bootstrap3': {
        'mezzanine_integration': {
            'form_template_choices': [
                ('fobi/bootstrap3_extras/view_embed_form.html',
                 "Custom bootstrap3 embed form view template"),
            ],
            'success_page_template_choices': [
                ('fobi/bootstrap3_extras/embed_form_submitted.html',
                 "Custom bootstrap3 embed form entry submitted template"),
            ],
        },
    },
    'foundation5': {
        'mezzanine_integration': {
            'form_template_choices': [
                ('fobi/foundation5_extras/view_embed_form.html',
                 "Custom foundation5 embed form view template"),
            ],
            'success_page_template_choices': [
                ('fobi/foundation5_extras/embed_form_submitted.html',
                 "Custom foundation5 embed form entry submitted template"),
            ],
        },
    },
}
```

39.6.4 Usage

39.6.4.1 The fobi page model

The `fobi.contrib.apps.mezzanine_integration.models.FobiFormPage` consists of the following fields:

- Form: The form to be used.
- Form template name: Template to be used to render the embed form.
- Hide form title: If checked, no form title would be shown.
- Form title: Overrides the standard form title.
- Submit button text: Overrides the default submit button text.
- Success page template name: Template to be used to render the embed form-sent event.
- Hide success page title: If checked, no form-sent title would be shown.
- Success page title: Overrides the form-sent title.
- Success page text: Overrides the form-sent text.

39.6.4.2 Steps described

1. If you use the mezzanine `example` project, to start go to the `http://localhost:8003/fobi/` URL and create a form.
2. Then go to `http://localhost:8003/admin/pages/page/` and add a new *Fobi form* page.
3. Choose the form and optionally - override the form settings and then save the page.
4. See the page in the front-end.

39.7 fobi.contrib.apps.wagtail_integration

A django-fobi integration with Wagtail.

39.7.1 Prerequisites

Tested with Wagtail 1.10.1 only. Might work on earlier (or later) versions as well.

39.7.2 Installation

39.7.2.1 Versions

See the requirements files:

- Wagtail 1.10 requirements.

39.7.2.2 your_project/settings.py

See the `example` settings file.

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    # ... standard wagtail apps

    # ... standard django-fobi apps

    'fobi.contrib.apps.wagtail_integration', # Wagtail integration app
]
```

39.7.2.3 your_project/your_app/models.py

If existing `fobi.contrib.apps.wagtail_integration.models.FobiFormPage` model does not fit your needs and you want to extend, there's a `fobi.contrib.apps.wagtail_integration.abstract.AbstractFobiFormPage` abstract model which you can extend. If so, remove the `fobi.contrib.apps.wagtail_integration` from `INSTALLED_APPS` and add path to the app with your customised `FobiFormPage` model.

```
from fobi.contrib.apps.wagtail_integration.models import AbstractFobiFormPage

class FobiFormPage(AbstractFobiFormPage):
    """Fobi form page."""

    # ... customise your form page further
```

And then:

```
INSTALLED_APPS = [
    # ... standard wagtail apps

    # ... standard django-fobi apps

    'your_app', # Customised `FobiFormPage` model app
]
```

39.7.2.4 Information for developers

39.7.2.4.1 Template rendering

The embed Wagtail page is rendered with use of two theme templates:

- `view_embed_form_entry_ajax_template`: Used for rendering the form.
- `embed_form_entry_submitted_ajax_template`: Used for rendering the form sent event.

39.7.2.5 Using custom templates for rendering the widget

In the widget, you can specify a template which you want to be used for rendering the form or the form-sent event.

Example:

```
FOBI_WAGTAIL_INTEGRATION_FORM_TEMPLATE_CHOICES = (
    ("yourapp/custom_view_embed_form_v1.html",
     "Custom embed form view template #1"),
    ("yourapp/custom_view_embed_form_v2.html",
     "Custom embed form view template #2"),
)
```

Same goes for form-sent templates.

```
FOBI_WAGTAIL_INTEGRATION_SUCCESS_PAGE_TEMPLATE_CHOICES = (
    ("yourapp/custom_embed_form_submitted_v1.html",
     "Custom form-sent template #1"),
    ("yourapp/custom_embed_form_submitted_v2.html",
     "Custom form-sent template #2"),
)
```

Registering a template in the `FORM_TEMPLATE_CHOICES` makes it available for all the themes. If you rather want to use different custom templates for different themes, use the `FOBI_CUSTOM_THEME_DATA` as shown in the example below.


```
FOBI_CUSTOM_THEME_DATA = {
    'bootstrap3': {
        'wagtail_integration': {
            'form_template_choices': [
                ('fobi/bootstrap3_extras/view_embed_form.html',
                 "Custom form view (non-partial) template"),
            ],
            'success_page_template_choices': [
                ('fobi/bootstrap3_extras/embed_form_submitted.html',
                 "Custom form entry submitted (non-partial) template"),
            ],
        },
    },
    'foundation5': {
        'wagtail_integration': {
            'form_template_choices': [
                ('fobi/foundation5_extras/view_embed_form.html',
                 "Custom form view (non-partial) template"),
            ],
            'success_page_template_choices': [
                ('fobi/foundation5_extras/embed_form_submitted.html',
                 "Custom form entry submitted (non-partial) template"),
            ],
        },
    },
}
```

Disregard the name, both `view_embed_form.html` and `embed_form_submitted.html` files should be full (non-partial) HTML templates.

39.7.3 Usage

The `fobi.contrib.apps.wagtail_integration.models.FobiFormPage` consists of the following django-fobi specific fields (as well as of other Wagtail Page related fields):

- Form: The form to be used.
- Form template name: Template to be used to render the embed form.
- Hide form title: If checked, no form title would be shown.
- Form title: Overrides the standard form title.
- Submit button text: Overrides the default submit button text.
- Success page template name: Template to be used to render the embed form-sent event.
- Hide success page title: If checked, no form-sent title would be shown.
- Success page title: Overrides the form-sent title.
- Success page text: Overrides the form-sent text.

FORM ELEMENTS

40.1 fobi.contrib.plugins.form_elements.content.content_image

A Fobi Image form element plugin.

40.1.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_image` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'easy_thumbnails',  
    'fobi.contrib.plugins.form_elements.content.content_image',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_image.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_IMAGE_` to the desired variable name from the above mentioned defaults module.

40.2 fobi.contrib.plugins.form_elements.content.content_image_url

A Fobi ImageURL form element plugin.

40.2.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_image_url` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.content.content_image_url',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_image_url.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_IMAGE_URL_` to the desired variable name from the above mentioned defaults module.

40.3 fobi.contrib.plugins.form_elements.content.content_markdown

A Fobi Markdown form element plugin based on [remarkable](#) and [markdown](#).

40.3.1 Installation

- (1) Install `markdown`.

```
pip install markdown
```

- (2) Add `fobi.reusable.markdown_widget` to `INSTALLED_APPS` in `settings.py`.

```
INSTALLED_APPS = (  
    ...  
    'fobi.reusable.markdown_widget',  
    ...  
)
```

- (3) Add `fobi.contrib.plugins.form_elements.content.content_markdown` to `INSTALLED_APPS` in `settings.py`.

```
INSTALLED_APPS = (  
    ...  
    'fobi.contrib.plugins.form_elements.content.content_markdown',  
    ...  
)
```

- (4) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (5) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.4 fobi.contrib.plugins.form_elements.content.content_richtext

A Fobi Rich text form element plugin based on [CKEditor](#) and [django-ckeditor](#).

40.4.1 Installation

- (1) Install `django-ckeditor`.

```
pip install django-ckeditor
```

- (2) Add `ckeditor` to `INSTALLED_APPS` in `settings.py`.

```
INSTALLED_APPS = (
    ...
    'ckeditor',
    ...
)
```

- (3) Add `fobi.contrib.plugins.form_elements.content.content_richtext` to `INSTALLED_APPS` in `settings.py`.

```
INSTALLED_APPS = (
    ...
    'fobi.contrib.plugins.form_elements.content.content_richtext',
    ...
)
```

- (4) Add `fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget` to `INSTALLED_APPS` in `settings.py` (if you're using `bootstrap3` theme). If you're using another theme, add correspondent widget specific to the active theme.

```
INSTALLED_APPS = (
    ...
    'fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_
↪bootstrap3_widget',
    ...
)
```

- (5) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (6) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.4.2 Controlling HTML tags and attributes

- (1) Install bleach.

```
pip install bleach
```

- (2) Specify `FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_TAGS`, `FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_ATTRIBUTES` and `FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_STYLES` in `settings.py`. The default values come from bleach:

```
FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_TAGS = [
    'a',
    'abbr',
    'acronym',
    'b',
    'blockquote',
    'code',
    'em',
    'i',
    'li',
    'ol',
    'strong',
    'ul',
]

FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_ATTRIBUTES = {
    'a': ['href', 'title'],
    'abbr': ['title'],
    'acronym': ['title'],
}

FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_STYLES = []
```

40.5 fobi.contrib.plugins.form_elements.content.content_text

A Fobi Text form element plugin.

40.5.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.content.content_text',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_text.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_TEXT_` to the desired variable name from the above mentioned defaults module.

By default the content of the text field is stripped using either the awesome [bleach](#) library or if bleach is not installed just Django's own `strip_tags` function. To configure the strip (bleach only) behaviour, two settings are introduced:

```
- ALLOWED_TAGS:
- ALLOWED_ATTRIBUTES:
```

The default values are:

```
ALLOWED_TAGS = [
    'a',
    'abbr',
    'acronym',
    'b',
    'blockquote',
    'code',
    'em',
    'i',
    'li',
    'ol',
    'strong',
    'ul',
]

ALLOWED_ATTRIBUTES = {
    'a': ['href', 'title'],
    'abbr': ['title'],
    'acronym': ['title'],
}
```

40.6 fobi.contrib.plugins.form_elements.content.content_video

A Fobi Video form element plugin.

40.6.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_video` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.content.content_video',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_video.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_VIDEO_` to the desired variable name from the above mentioned defaults module.

40.7 fobi.contrib.plugins.form_elements.fields.boolean

A Fobi Boolean form field plugin. Makes use of the `django.forms.fields.BooleanField`.

40.7.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.boolean',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.8 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple

A Fobi Select Multiple form field plugin. Makes use of the `django.forms.fields.MultipleChoiceField` and `django.forms.widgets.CheckboxSelectMultiple`.

40.8.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple',  
    # ...  
)
```

- (2) In the terminal type:


```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default, the submitted form value of *checkbox_select_multiple* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

- “val”: *value* (example: “alpha”).
- “repr” (default): *label* (example: “Alpha”).
- “mix”: *value (label)* (example: “Alpha (alpha)”).

Simply set the `FOBI_FORM_ELEMENT_CHECKBOX_SELECT_MULTIPLE_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.8.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<ul id="id_NAME_OF_THE_ELEMENT">
  <li>
    <label for="id_NAME_OF_THE_ELEMENT_0">
      <input class="form-control" id="id_NAME_OF_THE_ELEMENT_0" name="NAME_OF_THE_ELEMENT"
↪ type="checkbox" value="1" /> 1
    </label>
  </li>
  <li>
    <label for="id_NAME_OF_THE_ELEMENT_1">
      <input class="form-control" id="id_NAME_OF_THE_ELEMENT_1" name="NAME_OF_THE_ELEMENT"
↪ type="checkbox" value="2" /> 2
    </label>
  </li>
  <li>
```

(continues on next page)

(continued from previous page)

```
<label for="id_NAME_OF_THE_ELEMENT_2">
  <input class="form-control" id="id_NAME_OF_THE_ELEMENT_2" name="NAME_OF_THE_ELEMENT
↪" type="checkbox" value="alpha" /> Alpha
</label>
</li>
<li>
  <label for="id_NAME_OF_THE_ELEMENT_3">
    <input class="form-control" id="id_NAME_OF_THE_ELEMENT_3" name="NAME_OF_THE_ELEMENT
↪" type="checkbox" value="beta" /> Beta
  </label>
</li>
<li>
  <label for="id_NAME_OF_THE_ELEMENT_4">
    <input class="form-control" id="id_NAME_OF_THE_ELEMENT_4" name="NAME_OF_THE_ELEMENT
↪" type="checkbox" value="omega" /> omega
  </label>
</li>
</ul>
```

40.9 fobi.contrib.plugins.form_elements.fields.date

A Fobi Date form field plugin. Makes use of the `django.forms.fields.DateField` and `django.forms.widgets.DateInput`.

40.9.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.date` to the `INSTALLED_APPS` in your settings.py.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.date',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.10 fobi.contrib.plugins.form_elements.fields.date_drop_down

A Fobi Birthday form field plugin. Makes use of the `django.forms.fields.DateField` and `django.forms.extras.widgets.SelectDateWidget`.

40.10.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.date_drop_down` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.date_drop_down',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.11 fobi.contrib.plugins.form_elements.fields.datetime

A Fobi DateTime form field plugin. Makes use of the `django.forms.fields.DateTimeField` and `django.forms.widgets.DateTimeInput`.

40.11.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.datetime` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.datetime',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.12 fobi.contrib.plugins.form_elements.fields.decimal

A Fobi Decimal form field plugin. Makes use of the `django.forms.fields.DecimalField` and `django.forms.widgets.NumberInput` (falling back to `django.forms.widgets.TextInput` for older Django versions).

40.12.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.decimal` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.decimal',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.13 fobi.contrib.plugins.form_elements.fields.duration

A Fobi Select form field plugin. Makes use of the `django.forms.fields.DurationField`.

40.13.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.duration` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.duration',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.13.2 Usage

Next to standard field attributes, duration plugin accepts the following attributes (all integers):

- weeks:
- days:
- hours
- minutes:
- seconds:

All being used to form a `datetime.timedelta` value with the arguments mentioned above.

Initial value should be a properly formatted string.

```
3 days, 23:10:53
09:30:00
```

40.14 fobi.contrib.plugins.form_elements.fields.email

A Fobi Email form field plugin. Makes use of the `django.forms.fields.EmailField` and `django.forms.widgets.TextInput`.

40.14.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.email` to the `INSTALLED_APPS` in your settings.py.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.email',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to True.

40.15 fobi.contrib.plugins.form_elements.fields.file

A Fobi File form field plugin. Makes use of the `django.forms.fields.FileField` and `django.forms.widgets.ClearableFileInput`.

40.15.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.file` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.file',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default uploaded files are stored in the “fobi_plugins/file” directory of the media root. If you want to change the directory location, set the `FOBI_PLUGIN_FIELDS_FILE_FILES_UPLOAD_DIR` value to the desired (relative) path.
- (5) You may optionally restrict uploaded files extensions by specifying the `allowed_extensions` field in the plugin.

40.16 fobi.contrib.plugins.form_elements.fields.float

A Fobi Integer form field plugin. Makes use of the `django.forms.fields.FloatField` and `django.forms.widgets.NumberInput` (falling back to `django.forms.widgets.TextInput` for older Django versions).

40.16.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.float` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.float',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.17 fobi.contrib.plugins.form_elements.fields.hidden

A Fobi Hidden form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.HiddenInput`.

40.17.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.hidden` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.hidden',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.18 fobi.contrib.plugins.form_elements.fields.input

A generic input form field plugin. Makes use of the `django.forms.fields.Field` and `django.forms.widgets.Input`. Comes with a lot of options you likely won't use every day.

The full list of supported HTML properties is listed below:

- `autocomplete`
- `autofocus`
- `disabled`
- `list`
- `max`
- `min`
- `multiple`
- `pattern`
- `placeholder`
- `readonly`
- `step`
- `type`

See w3schools.com for further explanations.

40.18.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.input` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.input',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.19 fobi.contrib.plugins.form_elements.fields.integer

A Fobi Integer form field plugin. Makes use of the `django.forms.fields.IntegerField` and `django.forms.widgets.NumberInput` (falling back to `django.forms.widgets.TextInput` for older Django versions).

40.19.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.integer` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.integer',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.20 fobi.contrib.plugins.form_elements.fields.ip_address

A Fobi Text form field plugin. Makes use of the `django.forms.fields.GenericIPAddressField` and `django.forms.widgets.TextInput`.

40.20.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.ip_address` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.ip_address',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.21 fobi.contrib.plugins.form_elements.fields.null_boolean

A Fobi NullBoolean form field plugin. Makes use of the `django.forms.fields.NullBooleanField` and `django.forms.widgets.NullBooleanSelect`.

40.21.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.null_boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.null_boolean',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.22 fobi.contrib.plugins.form_elements.fields.password

A Fobi Password form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.PasswordInput`.

40.22.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.password` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.password',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.23 fobi.contrib.plugins.form_elements.fields.radio

A Fobi Radio form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.RadioSelect`.

40.23.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.radio` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.radio',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default, the submitted form value of *radio* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[  
    ('alpha', 'Alpha'),  
    ('beta', 'Beta'),  
    ('gamma', 'Gamma'),  
]
```

- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").

Simply set the `FOBI_FORM_ELEMENT_RADIO_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.23.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="alpha">Alpha</option>
  <option value="beta">Beta</option>
  <option value="omega">omega</option>
</select>
```

40.24 fobi.contrib.plugins.form_elements.fields.range_select

A Fobi RangeSelect form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.Select`.

40.24.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.range_select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.range_select',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

(4) Ranges are specified within the given min/max values. The default values are:

```
- INITIAL: 50
- INITIAL_MAX_VALUE: 100
- INITIAL_MIN_VALUE: 0
- MIN_VALUE: 0
- MAX_VALUE: 100
- STEP: 1
```

However, you can override each of them in the settings of your project by prefixing correspondent names with `FOBI_FORM_ELEMENT_RANGE_SELECT_`:

```
- FOBI_FORM_ELEMENT_RANGE_SELECT_INITIAL
- FOBI_FORM_ELEMENT_RANGE_SELECT_INITIAL_MAX_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_INITIAL_MIN_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_MIN_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_MAX_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_STEP
```

(5) By default, the submitted form value of *range_select* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_RANGE_SELECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.25 fobi.contrib.plugins.form_elements.fields.regex

A Fobi Text form field plugin. Makes use of the `django.forms.fields.RegexField` and `django.forms.widgets.TextInput`.

40.25.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.regex` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.regex',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.26 fobi.contrib.plugins.form_elements.fields.select

A Fobi Select form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.Select`.

40.26.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.select',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default, the submitted form value of *select* elements is *label* (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.26.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="alpha">Alpha</option>
  <option value="beta">Beta</option>
  <option value="omega">omega</option>
</select>
```

40.27 fobi.contrib.plugins.form_elements.fields.select_model_object

A Fobi Select Model Object form field plugin. Makes use of the `django.forms.models.ModelChoiceField` and `django.forms.widgets.Select`.

40.27.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_model_object` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select_model_object',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_model_object.defaults.IGNORED_MODELS`. If necessary, override it in your *settings* as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MODEL_OBJECT_IGNORED_MODELS = [
    'auth.User',
    'auth.Group',
]
```

- (5) By default, the submitted form value of *select_model_object* elements is *app_label.model_name.object_pk.object_repr*. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MODEL_OBJECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.28 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object

A Fobi Select MPTT Model Object form field plugin. Makes use of the `mptt.fields.TreeNodeChoiceField` and `django.forms.widgets.Select`.

40.28.1 Installation

40.28.1.1 Install *django-mptt*

Taken from *django-mptt* [Getting started](#).

1. Download *django-mptt* using pip by running:

```
pip install django-mptt
```

2. Add *mptt* to the `INSTALLED_APPS` in your `settings.py`.

40.28.1.2 Install *select_mptt_model_object* plugin

- (1) Add `mptt` and `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'mptt',  
    'fobi.contrib.plugins.form_elements.fields.select_mptt_model_object',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults.IGNORED_MODELS`. If necessary, override it in your *settings* as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MPTT_MODEL_OBJECT_IGNORED_MODELS = [  
    'auth.User',  
    'auth.Group',  
]
```

- (5) By default, the submitted form value of *select_mptt_model_object* elements is `app_label.model_name.object_pk.object_repr`. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.  
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).  
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MPTT_MODEL_OBJECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.29 fobi.contrib.plugins.form_elements.fields.select_multiple

A Fobi Select Multiple form field plugin. Makes use of the `django.forms.fields.MultipleChoiceField` and `django.forms.widgets.SelectMultiple`.

40.29.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select_multiple',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default, the submitted form value of *select_multiple* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.29.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1
2
alpha, Alpha
```

(continues on next page)

(continued from previous page)

```
beta, Beta
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="alpha">Alpha</option>
  <option value="beta">Beta</option>
  <option value="omega">omega</option>
</select>
```

40.30 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects

A Fobi Select Multiple Model Objects form field plugin. Makes use of the `django.forms.models.ModelMultipleChoiceField` and `django.forms.widgets.SelectMultiple`.

40.30.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults.IGNORED_MODELS`. If necessary, override it in your *settings* as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MODEL_OBJECTS_IGNORED_MODELS = [
    'auth.User',
    'auth.Group',
]
```

- (5) By default, the submitted form value of *select_multiple_model_objects* elements is *app_label.model_name.object_pk.object_repr*. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MODEL_OBJECTS_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.31 fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects

A Fobi Select Multiple MPTT Model Objects form field plugin. Makes use of the `mptt.forms.TreeNodeMultipleChoiceField` and `django.forms.widgets.SelectMultiple`.

40.31.1 Installation

40.31.1.1 Install *django-mptt*

Taken from `django-mptt` [Getting started](#).

1. Download `django-mptt` using pip by running:

```
pip install django-mptt
```

2. Add `mptt` to the `INSTALLED_APPS` in your `settings.py`.

40.31.1.2 Install *select_multiple_mptt_model_objects* plugin

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'mptt',
    'fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults.IGNORED_MODELS`. If necessary, override it in your `settings` as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MPTT_MODEL_OBJECTS_IGNORED_MODELS = [
    'auth.User',
    'auth.Group',
]
```

- (5) By default, the submitted form value of *select_multiple_mptt_model_objects* elements is *app_label.model_name.object_pk.object_repr*. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MPTT_MODEL_OBJECTS_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.32 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max

A Fobi Select Multiple form field plugin with max choices. Makes use of the `django.forms.widgets.SelectMultiple`.

40.32.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max` to the `INSTALLED_APPS` in your settings.py.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select_multiple_with_max',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to True.
- (4) By default, the submitted form value of *select_multiple_with_max* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_WITH_MAX_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.32.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair. If you enter an integer in the ‘max_choices’ field, the user can choose only <max_choices> or less choices.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="alpha">Alpha</option>
  <option value="beta">Beta</option>
  <option value="omega">omega</option>
</select>
```

40.33 fobi.contrib.plugins.form_elements.fields.slider

A Fobi Percentage form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.Select`.

40.33.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.slider` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.slider',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Ranges are specified within the given min/max values. The default values are:

```
- INITIAL: 50
- INITIAL_MAX_VALUE: 100
- INITIAL_MIN_VALUE: 0
- MIN_VALUE: 0
```

(continues on next page)

(continued from previous page)

```
- MAX_VALUE: 100
- STEP: 1
```

However, you can override each of them in the settings of your project by prefixing correspondent names with `FOBI_FORM_ELEMENT_SLIDER_`:

```
- FOBI_FORM_ELEMENT_SLIDER_INITIAL
- FOBI_FORM_ELEMENT_SLIDER_INITIAL_MAX_VALUE
- FOBI_FORM_ELEMENT_SLIDER_INITIAL_MIN_VALUE
- FOBI_FORM_ELEMENT_SLIDER_MIN_VALUE
- FOBI_FORM_ELEMENT_SLIDER_MAX_VALUE
- FOBI_FORM_ELEMENT_SLIDER_STEP
```

- (5) By default, the submitted form value of *slider* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SLIDER_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.34 fobi.contrib.plugins.form_elements.fields.slug

A Fobi Text form field plugin. Makes use of the `django.forms.fields.SlugField` and `django.forms.widgets.TextInput`.

40.34.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.slug` to the `INSTALLED_APPS` in your settings.py.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.slug',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.35 fobi.contrib.plugins.form_elements.fields.text

A Fobi Text form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.TextInput`.

40.35.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.text',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.36 fobi.contrib.plugins.form_elements.fields.textarea

A Fobi Textarea form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.Textarea`.

40.36.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.textarea` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.textarea',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.37 fobi.contrib.plugins.form_elements.fields.time

A Fobi DateTime form field plugin. Makes use of the `django.forms.fields.TimeField` and `django.forms.widgets.TextInput`.

40.37.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.time` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.time',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.38 fobi.contrib.plugins.form_elements.fields.url

A Fobi URL form field plugin. Makes use of the `django.forms.fields.URLField` and `django.forms.widgets.URLInput` falling back to `django.forms.widgets.TextInput` for older Django versions.

40.38.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.url` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.url',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.39 fobi.contrib.plugins.form_elements.security.captcha

A **CAPTCHA** form field plugin. Makes use of the [django-simple-captcha](#).

40.39.1 Prerequisites

You will need `libfreetype6`, otherwise `django-simple-captcha` won't work.

```
sudo apt-get install libfreetype6-dev
```

40.39.2 Installation

40.39.2.1 Install *django-simple-captcha*

Taken from [django-simple-captcha installation instructions](#).

- (1) Download `django-simple-captcha` using `pip` by running:

```
pip install django-simple-captcha
```

- (2) Add `captcha` to the `INSTALLED_APPS` in your `settings.py`.
- (3) Run `python manage.py migrate`.
- (4) Add an entry to your `urls.py`:

```
urlpatterns += [
    url(r'^captcha/', include('captcha.urls')),
]
```

40.39.2.2 Install *fobi* Captcha plugin

- (1) Add `fobi.contrib.plugins.form_elements.security.captcha` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.security.captcha',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) In order to have nicer text input widget, add the following line to your settings (if you're using `bootstrap3` theme):

```
CAPTCHA_TEXT_FIELD_TEMPLATE = 'bootstrap3/captcha/text_field.html'
```

For `foundation5` theme add the following line:

```
CAPTCHA_TEXT_FIELD_TEMPLATE = 'foundation5/captcha/text_field.html'
```

40.39.3 Troubleshooting and usage limitations

40.39.3.1 In combination with other captcha solutions

At the moment, you can't use both CAPTCHA (`fobi.contrib.plugins.form_elements.security.captcha`) and ReCAPTCHA (`fobi.contrib.plugins.form_elements.security.recaptcha`) plugins alongside due to app name collision of the `django-simple-captcha` and `django-recaptcha` packages.

40.39.4 Usage

Note, that unlike most of the other form element plugins, default value for the `required` attribute is `True`, which makes the Captcha obligatory. Although you could still set it to `False`, it does not make much sense to do so.

40.40 `fobi.contrib.plugins.form_elements.security.honeypot`

A [Honeypot](#) form field plugin. Just another anti-spam technique.

40.40.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.security.honeypot` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.security.honeypot',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.41 `fobi.contrib.plugins.form_elements.security.invisible_recaptcha`

A [Invisible reCAPTCHA](#) form field plugin. Just another anti-spam technique.

40.41.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.security.invisible_recaptcha` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.security.invisible_recaptcha',
    'fobi.contrib.themes.bootstrap3.widgets.form_elements.invisible_recaptcha_
    ↪bootstrap3_widget',
    # ...
)
```

- (2) Enable Invisible reCAPTCHA for your site and put the site key and site secret keys in the settings:

```
FOBI_PLUGIN_INVISIBLE_RECAPTCHA_SITE_KEY = 'your-site-key'
FOBI_PLUGIN_INVISIBLE_RECAPTCHA_SITE_SECRET = 'your-site-secret'
```

- (3) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (4) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.42 fobi.contrib.plugins.form_elements.security.recaptcha

A [ReCAPTCHA](#) form field plugin. Makes use of the [django-recaptcha](#).

40.42.1 Installation

40.42.1.1 Install *django-recaptcha*

- (1) Download `django-recaptcha` using `pip` by running:

```
pip install django-recaptcha
```

- (2) Add `captcha` to the `INSTALLED_APPS` in your `settings.py`.
- (3) Run `python manage.py migrate`.

40.42.1.2 Install *fobi* ReCAPTCHA plugin

- (1) Add `fobi.contrib.plugins.form_elements.security.recaptcha` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.security.recaptcha',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

(4) Specify the following ReCAPTCHA credentials in your settings:

```
- ``RECAPTCHA_PUBLIC_KEY``  
- ``RECAPTCHA_PRIVATE_KEY``
```

For testing purposes obtain the keys [here](#):

40.42.2 Troubleshooting and usage limitations

40.42.2.1 In combination with other captcha solutions

At the moment, you can't use both CAPTCHA (`fobi.contrib.plugins.form_elements.security.captcha`) and ReCAPTCHA (`fobi.contrib.plugins.form_elements.security.recaptcha`) plugins alongside due to app name collision of the `django-simple-captcha` and `django-recaptcha` packages.

If you happen to see errors like “Input error: k: Format of site key was invalid”, make sure to have defined (and filled in properly) the `RECAPTCHA_PUBLIC_KEY` and `RECAPTCHA_PRIVATE_KEY` in your `settings.py`. See the [following](#) thread for more information.

40.42.3 Usage

Note, that unlike most of the other form element plugins, default value for the `required` attribute is `True`, which makes the ReCaptcha obligatory. Although you could still set it to `False`, it does not make much sense to do so.

40.43 fobi.contrib.plugins.form_elements.test.dummy

A Fobi Dummy form element plugin. Created for testing purposes.

40.43.1 Installation

(1) Add `fobi.contrib.plugins.form_elements.test.dummy` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.test.dummy',  
    # ...  
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

FORM HANDLERS

41.1 fobi.contrib.plugins.form_handlers.db_store

A Fobi Database Store form-/wizard- handler plugin. Saves submitted form data into the SavedFormDataEntry/SavedFormWizardDataEntry models.

41.1.1 Dependencies

The `xlwt` package is required (optional) for XLS export. If not present, export format falls back to CSV.

41.1.2 Installation

- (1) Add `fobi.contrib.plugins.form_handlers.db_store` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_handlers.db_store',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py migrate  
  
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Add `db_store` form handler plugin URLs to the `urls.py` of your project.

```
urlpatterns = [  
    # DB Store plugin URLs  
    url(r'^fobi/plugins/form-handlers/db-store/',  
        include('fobi.contrib.plugins.form_handlers.db_store.urls')),  
]
```

For form wizards do:

```
urlpatterns = [  
    # DB Store plugin URLs  
    url(r'^fobi/plugins/form-wizard-handlers/db-store/',  
        include('fobi.contrib.plugins.form_handlers.db_store.urls.'  
                'form_wizard_handlers')),  
]
```

41.2 fobi.contrib.plugins.form_handlers.http_repost

A Fobi HTTP Repost form handler plugin. Submits the form data as is to the given endpoint.

41.2.1 Installation

- (1) Add `fobi.contrib.plugins.form_handlers.http_repost` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_handlers.http_repost',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

41.3 fobi.contrib.plugins.form_handlers.mail

A Fobi Mail form handler plugin. Submits the form data by email to the specified email address.

41.3.1 Installation

- (1) Add `fobi.contrib.plugins.form_handlers.mail` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_handlers.mail',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

41.4 fobi.contrib.plugins.form_handlers.mail_sender

A Fobi Mail form handler plugin. Submits the form data by email to the sender (submitter) of the form.

41.4.1 Installation

- (1) Add `fobi.contrib.plugins.form_handlers.mail_sender` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_handlers.mail_sender',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

41.4.2 Usage

- (1) Add the *Mail the sender* form handler to the form.
- (2) In the value for *Form field name to email* fill in the form field name which shall be used for sending an email to. For instance, if your original form consisted of *name*, *email*, *comment* fields, you should put *email* as a value for *Form field name to email* field.

FORM IMPORTERS

42.1 mailchimp_importer

A `django-fobi` integration with MailChimp.

This plugin makes it possible to import a form from a MailChimp list. A typical list URL would be [this](#). In the listing you would see list names and *Stats* at the right corner. If you click on it you would see the *Settings* link. Follow it and scroll to the bottom for the unique id for your list. Now, if you have been successfully authenticated to the MailChimp API using your `API_KEY`, you could call the `lists.merge_vars` method for getting the form. `API_KEY` could be obtained from the MailChimp in the [Account API](#).

For additional information on MailChimp import see the following [article](#).

42.1.1 Prerequisites

Python wrapper for the Mailchimp:

```
pip install mailchimp
```

If you are using Django 1.8 or greater, you would need *django-formtools* package as well:

```
pip install django-formtools
```

42.1.2 Installation

42.1.2.1 your_project/settings.py

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    'fobi.contrib.plugins.form_importers.mailchimp_importer',
]
```

42.1.3 How it works

Assuming that you have configured the *mailchimp_importer* plugin properly and have the Django running locally on port 8000, accessing the following URL would bring you to the MailChimp form import wizard.

- <http://localhost:8000/en/fobi/forms/importer/mailchimp/>

On the first step you would be asked to provide your API_KEY, which is used to authenticate to the MailChimp in order to fetch your list- and form- information. The key isn't stored/saved/remembered. Next time you want to import a form from the same account, you would have to provide it again.

42.1.4 Development status

This part of code is alpha, which means it experimental and needs improvements.

See the [TODOS](#) for the full list of planned-, pending- in-development- or to-be-implemented features.

If you want to improve it or did make it working, please, make a pull request.

43.1 fobi.contrib.themes.bootstrap3

A django-fobi Bootstrap 3 theme. Based on the ??? template.

43.1.1 Installation

- (1) Add `fobi.contrib.themes.bootstrap3` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.themes.bootstrap3',  
    # ...  
)
```

- (2) Specify `bootstrap3` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'bootstrap3'
```

43.2 fobi.contrib.themes.djangocms_admin_style_theme

A django-fobi theme in a style of `djangocms-admin-style` admin. Relies on `djangocms-admin-style` package and some jQuery UI only.

jQuery UI “Smoothness” theme comes from [here](#).

43.2.1 Installation

43.2.1.1 Install *djangocms-admin-style*

See the original [installation instructions](#).

- (1) Install the `djangocms-admin-style` package.

```
pip install djangocms-admin-style
```

- (2) Add `djangocms_admin_style` to your `INSTALLED_APPS` just before `django.contrib.admin`.

43.2.1.2 Install `fobi.contrib.themes.djangocms_admin_style_theme` theme

- (1) Add `fobi.contrib.themes.djangocms_admin_style_theme` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.themes.djangocms_admin_style_theme',  
    # ...  
)
```

- (2) Specify `djangocms_admin_style_theme` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'djangocms_admin_style_theme'
```

43.3 `fobi.contrib.themes.foundation5`

A django-fobi Foundation 5 theme. Based on the ??? template, but entire JS and CSS are taken from Foundation 5 version 5.4.0. The [following](#) icon set was used.

43.3.1 Installation

- (1) Add `fobi.contrib.themes.foundation5` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.themes.foundation5',  
    # ...  
)
```

- (2) Specify `foundation5` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'foundation5'
```

43.4 `fobi.contrib.themes.simple`

A django-fobi theme in a style of Django admin. Relies on Django-admin and some jQuery UI only. jQuery UI “Django” theme comes from [here](#).

43.4.1 Installation

- (1) Add `fobi.contrib.themes.simple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.themes.simple',  
    # ...  
)
```

- (2) Specify `simple` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'simple'
```


FOBI.INTEGRATION

Sub-package for easy integration of fobi with third-party apps (especially content management systems).

CHAPTER
FORTYFIVE

CONCEPTS

TEMPLATES

46.1 Reusable

Home for reusable apps that live in fobi codebase, but might be migrated elsewhere some day.

46.1.1 Markdown widget

Markdown widget for Django based on [remarkable](#).

FOBI.WIZARD

Persistent and working form-wizards.

Contents:

48.1 Quick start

Tutorial for very quick start with `django-fobi`. Consists of several parts listed below:

- Part 1: Standard Django installation
- Part 2: Integration **with** DjangoCMS (coming soon)

48.1.1 Part 1: standard Django installation

Example project code available [here](#).

48.1.1.1 Installation and configuration

48.1.1.1.1 Install the package in your environment.

```
pip install django-fobi
```

48.1.1.1.2 INSTALLED_APPS

Add `fobi` core and the plugins to the `INSTALLED_APPS` of the your `settings` module.

1. The core.

```
'fobi',
```

2. The preferred theme. Bootstrap 3 theme is the default. If you have chosen a different theme, update the value of `FOBI_DEFAULT_THEME` accordingly.

```
'fobi.contrib.themes.bootstrap3',
```

3. The form field plugins. Plugins are like blocks. You are recommended to have them all installed. Note, that the following plugins do not have additional dependencies, while some others (like `fobi.contrib.plugins.form_elements.security.captcha` or `fobi.contrib.plugins.form_elements.security.recaptcha`) would require additional packages to be installed. If so, make sure to have installed and configured those dependencies prior adding the dependant add-ons to the `settings` module.

```
'fobi.contrib.plugins.form_elements.fields.boolean',
'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple',
'fobi.contrib.plugins.form_elements.fields.date',
'fobi.contrib.plugins.form_elements.fields.date_drop_down',
'fobi.contrib.plugins.form_elements.fields.datetime',
'fobi.contrib.plugins.form_elements.fields.decimal',
'fobi.contrib.plugins.form_elements.fields.email',
'fobi.contrib.plugins.form_elements.fields.file',
'fobi.contrib.plugins.form_elements.fields.float',
'fobi.contrib.plugins.form_elements.fields.hidden',
'fobi.contrib.plugins.form_elements.fields.input',
'fobi.contrib.plugins.form_elements.fields.integer',
'fobi.contrib.plugins.form_elements.fields.ip_address',
'fobi.contrib.plugins.form_elements.fields.null_boolean',
'fobi.contrib.plugins.form_elements.fields.password',
'fobi.contrib.plugins.form_elements.fields.radio',
'fobi.contrib.plugins.form_elements.fields.regex',
'fobi.contrib.plugins.form_elements.fields.select',
'fobi.contrib.plugins.form_elements.fields.select_model_object',
'fobi.contrib.plugins.form_elements.fields.select_multiple',
'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects',
'fobi.contrib.plugins.form_elements.fields.slug',
'fobi.contrib.plugins.form_elements.fields.text',
'fobi.contrib.plugins.form_elements.fields.textarea',
'fobi.contrib.plugins.form_elements.fields.time',
'fobi.contrib.plugins.form_elements.fields.url',
```

4. The presentational form elements (images, texts, videos).

```
'easy_thumbnails', # Required by `content_image` plugin
'fobi.contrib.plugins.form_elements.content.content_image',
'fobi.contrib.plugins.form_elements.content.content_image_url',
'fobi.contrib.plugins.form_elements.content.content_text',
'fobi.contrib.plugins.form_elements.content.content_video',
```

5. Form handlers. Note, that some of them may require database sync/migration.

```
'fobi.contrib.plugins.form_handlers.db_store',
'fobi.contrib.plugins.form_handlers.http_repost',
'fobi.contrib.plugins.form_handlers.mail',
'fobi.contrib.plugins.form_handlers.mail_sender',
```

Putting all together, you would have something like this.

```
INSTALLED_APPS = (
    # Used by fobi
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django.contrib.admin',
```

(continues on next page)

(continued from previous page)

```

# ...
# Core
'fobi',

# Theme
'fobi.contrib.themes.bootstrap3',

# Form field plugins
'fobi.contrib.plugins.form_elements.fields.boolean',
'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple',
'fobi.contrib.plugins.form_elements.fields.date',
'fobi.contrib.plugins.form_elements.fields.date_drop_down',
'fobi.contrib.plugins.form_elements.fields.datetime',
'fobi.contrib.plugins.form_elements.fields.decimal',
'fobi.contrib.plugins.form_elements.fields.email',
'fobi.contrib.plugins.form_elements.fields.file',
'fobi.contrib.plugins.form_elements.fields.float',
'fobi.contrib.plugins.form_elements.fields.hidden',
'fobi.contrib.plugins.form_elements.fields.input',
'fobi.contrib.plugins.form_elements.fields.integer',
'fobi.contrib.plugins.form_elements.fields.ip_address',
'fobi.contrib.plugins.form_elements.fields.null_boolean',
'fobi.contrib.plugins.form_elements.fields.password',
'fobi.contrib.plugins.form_elements.fields.radio',
'fobi.contrib.plugins.form_elements.fields.regex',
'fobi.contrib.plugins.form_elements.fields.select',
'fobi.contrib.plugins.form_elements.fields.select_model_object',
'fobi.contrib.plugins.form_elements.fields.select_multiple',
'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects',
'fobi.contrib.plugins.form_elements.fields.slug',
'fobi.contrib.plugins.form_elements.fields.text',
'fobi.contrib.plugins.form_elements.fields.textarea',
'fobi.contrib.plugins.form_elements.fields.time',
'fobi.contrib.plugins.form_elements.fields.url',

# Form element plugins
'easy_thumbnails', # Required by ``content_image`` plugin
'fobi.contrib.plugins.form_elements.content.content_image',
'fobi.contrib.plugins.form_elements.content.content_image_url',
'fobi.contrib.plugins.form_elements.content.content_text',
'fobi.contrib.plugins.form_elements.content.content_video',

# Form handlers
'fobi.contrib.plugins.form_handlers.db_store',
'fobi.contrib.plugins.form_handlers.http_repost',
'fobi.contrib.plugins.form_handlers.mail',
'fobi.contrib.plugins.form_handlers.mail_sender',

# ...
)

```

48.1.1.1.3 TEMPLATE_CONTEXT_PROCESSORS

Add `django.core.context_processors.request` and `fobi.context_processors.theme` to `TEMPLATES` of your settings module.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [(os.path.join('path', 'to', 'your', 'templates'))],
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                'fobi.context_processors.theme', # Important!
                'fobi.context_processors.dynamic_values', # Optional
            ],
            'loaders': [
                'django.template.loaders.filesystem.Loader',
                'django.template.loaders.app_directories.Loader',
                'admin_tools.template_loaders.Loader',
            ],
            'debug': DEBUG_TEMPLATE,
        },
    },
]
```

48.1.1.1.4 urlpatterns

Add the following line to `urlpatterns` of your `urls` module.

```
urlpatterns = [
    # ...

    # DB Store plugin URLs
    url(r'^fobi/plugins/form-handlers/db-store/',
        include('fobi.contrib.plugins.form_handlers.db_store.urls')),

    # View URLs
    url(r'^fobi/', include('fobi.urls.view')),

    # Edit URLs
    url(r'^fobi/', include('fobi.urls.edit')),

    # ...
]
```

48.1.1.1.5 Update the database

1. First you should migrate the database. Depending on your Django version and migration app, this step may vary. Typically as follows:

```
./manage.py migrate --fake-initial
```

2. Sync installed fobi plugins. Go to terminal and type the following command.

```
./manage.py fobi_sync_plugins
```

48.1.1.1.6 Specify the active theme

Specify the default theme in your `settings` module.

```
FOBI_DEFAULT_THEME = 'bootstrap3'
```

48.1.1.1.7 Permissions

fobi has been built with permissions in mind. Every single form element plugin or handler is permission based. If user hasn't been given permission to work with a form element or a form handler plugin, he won't be. If you want to switch the permission checks off, set the value of `FOBI_RESTRICT_PLUGIN_ACCESS` to `False` in your *settings* module.

```
FOBI_RESTRICT_PLUGIN_ACCESS = False
```

Otherwise, after having completed all the steps above, do log into the Django administration and assign the permissions (to certain user or a group) for every single form element or form handler plugin. Bulk assignments work as well.

- <http://yourdomain.com/admin/fobi/formelement/>
- <http://yourdomain.com/admin/fobi/formhandler/>

Also, make sure to have the Django model permissions set for following models:

- `fobi.models.FormEntry`
- `fobi.models.FormElementEntry`
- `fobi.models.FormHandlerEntry`
- `fobi.contrib.plugins.form_handlers.db_store.models.SavedFormDataEntry`

48.1.2 Part 2: Integration with DjangoCMS

Coming soon...

48.2 Release history and notes

Sequence based identifiers are used for versioning (schema follows below):

```
major.minor[.revision]
```

- It's always safe to upgrade within the same minor version (for example, from 0.3 to 0.3.4).
- Minor version changes might be backwards incompatible. Read the release notes carefully before upgrading (for example, when upgrading from 0.3.4 to 0.4).
- All backwards incompatible changes are mentioned in this document.

48.2.1 0.19.4

2022-08-08

- Minor fixes in docs.

48.2.2 0.19.3

2022-08-08

- Minor fixes in the *rich_text* plugin (*bleach* related).
- Minor fixes and clean up in docs.

48.2.3 0.19.2

2022-07-14

- Make it easier to get initial data for the *ViewFormEntry* view.
- Apply black and isort on entire code base.

48.2.4 0.19.1

2022-07-12

- Added class-based view for Dashboard.

48.2.5 0.19

2022-07-11

- Introduce class based views. Function based views are still supported and will be supported until at least 0.23. Migration to class based views is simple. Only your project's `urls.py` would change:

```
urlpatterns = [  
    # ...  
    url(r'^fobi/', include('fobi.urls.class_based.view')),  
    url(r'^fobi/', include('fobi.urls.class_based.edit')),
```

(continues on next page)

(continued from previous page)

```
# ...
]
```

To use function based views, simply replace the previous line with:

```
urlpatterns = [
    # ...
    url(r'^fobi/', include('fobi.urls.view')),
    url(r'^fobi/', include('fobi.urls.edit')),
    # ...
]
```

- Class-based permissions (work only in combination with class-based views).

Example:

```
from fobi.permissions.definitions import edit_form_entry_permissions
from fobi.permissions.generic import BasePermission
from fobi.permissions.helpers import (
    any_permission_required_func, login_required,
)

class EditFormEntryPermission(BasePermission):
    """Permission to edit form entries."""

    def has_permission(self, request, view) -> bool:
        return login_required(request) and any_permission_required_func(
            edit_form_entry_permissions
        )(request.user)

    def has_object_permission(self, request, view, obj) -> bool:
        return login_required(request) and any_permission_required_func(
            edit_form_entry_permissions
        )(request.user) and obj.user == request.user
```

48.2.6 0.18

2022-06-23

- Tested against Python 3.9, Django 3.2 and 4.0.

Note: Release dedicated to my dear son, Tigran, who turned 10 recently.

48.2.7 0.17.1

2021-01-25

Note: Release dedicated to defenders of Armenia and Artsakh (Nagorno Karabakh) and all the victims of Turkish and Azerbaijani aggression.

- Replace outdated *simplejson* with *json*.

48.2.8 0.17

2020-12-28

Note: Release dedicated to defenders of Armenia and Artsakh (Nagorno Karabakh) and all the victims of Turkish and Azerbaijani aggression.

- Deprecate support for the following Django versions: 1.11, 2.0 and 2.1.

48.2.9 0.16.4

2020-04-08

- Replace *admin_static* templatetag module with *static*, since the latter is available in Django 1.11, which is now the minimal supported version.

48.2.10 0.16.3

2020-04-04

- Fixes in *invisible_recaptcha* plugin. Treat empty `SITE_KEY` as error.

48.2.11 0.16.2

2020-04-03

- Conditional use of deprecated *admin_static* templatetag module.

48.2.12 0.16.1

2020-03-18

- Fix the `datetime` is not JSON serializable error.

48.2.13 0.16

2020-01-11

Note: This release drops support for Python 2.7 and 3.4 and sets the minimum requirement version for Django to 1.11. If you don't meet these requirements - do not upgrade.

- Add support for Django 3.0.
- Drop support Django 1.8, 1.9, 1.10. Minimal supported Django version is 1.11.
- Tested against Python 3.8.
- Drop Python 2.7 support.
- Drop Python 3.4 support.
- Tested with Django REST Framework 3.11.x. All older versions of Django REST Framework should still work, although they are no longer tested.
- Tested with FeinCMS 1.17.x. No code changes at all. All older versions of FeinCMS should still work, although they are no longer tested.

48.2.14 0.15

2019-06-20

Note: Although this release does not contain major backwards incompatible changes, some things you have had working in past might break.

- The following settings `FOBI_FAIL_ON_ERRORS_IN_FORM_ELEMENT_PLUGINS`, `FOBI_FAIL_ON_ERRORS_IN_FORM_HANDLER_PLUGINS`, `FOBI_FAIL_ON_MISSING_INTEGRATION_FORM_ELEMENT_PLUGINS` and `FOBI_FAIL_ON_MISSING_INTEGRATION_FORM_HANDLER_PLUGINS` used to be set to `False` by default (silencing exceptions). It's no longer so. Exceptions will cause errors from now on (you might get error 500). If you want old behaviour, set them to `False`.

48.2.15 0.14

2019-05-24

- Tested again Django 2.2 and Python 3.7.
- Use `ChromeDriver` (for browser tests).
- Wiping out (as stated many versions ago) support for Django < 1.8.
- Reactivate travis (although not all tests yet pass, while they do pass locally; to be solved in future).

48.2.16 0.13.10

2019-02-28

- Add *mail_sender* DRF integration plugin.

48.2.17 0.13.9

2019-02-28

Note: Release supported by [Goldmund, Wyldebeast & Wunderliebe](#).

- Add *mail_sender* form handler plugin.
- Upgrade test suite.

48.2.18 0.13.8

2019-01-07

Note: Release supported by [Goldmund, Wyldebeast & Wunderliebe](#).

- Make it easier to redirect to a new URL (on success) in integration apps.

48.2.19 0.13.7

2018-10-01

- Make fobi wagtail integration work with wagtail 2.

48.2.20 0.13.6

2018-08-16

- Completely wiping out `django-autoslug` in requirements (for now it has been replaced with `django-autoslug-iplweb`).

48.2.21 0.13.5

2018-08-12

- Minor fixes in code, tests and docs.

48.2.22 0.13.4

2018-08-11

- Django 2.1 support (initial).
- Upgrade test suite.

48.2.23 0.13.3

2018-07-10

- Minor fixes admin export of form entries (Postgres).

48.2.24 0.13.2

2018-03-12

- Choices are now required fields for *checkbox_select_multiple*, *radio*, *select*, *select_multiple* and *select_multiple_with_max* form elements.
- The *active_date_from*, *active_date_to*, *inactive_page_title*, *inactive_page_message* added to forms the import/export flow.

48.2.25 0.13.1

2018-02-28

- Minor fixes in recaptcha plugin.
- Add *max_length* to textarea plugin.

48.2.26 0.13

2018-02-25

Note: This release contains minor backwards incompatible changes. A number of new database fields have been added to the *FormEntry* model (*active_date_from*, *active_date_to*, *inactive_page_title*, *inactive_page_message*). You will need to migrate your database.

- Implement disabling forms based on dates. Note, that although the feature is landed into the core, contrib packages (such as Foundation 5 theme, Simple theme, Django CMS admin style theme, DRF integration app, FeinCMS integration app, Mezzanine integration app, DjangoCMS integration app and Wagtail app) are not yet updated to reflect these changes. It's a work in progress to be done in the nearest minor releases. Check the [issue #143](#) for state of the progress.

48.2.27 0.12.20

2018-02-24

- Minor Python 2 fixes.

48.2.28 0.12.19

2018-02-21

- Make it possible to sort on plugin name instead of uid.

48.2.29 0.12.18

2018-02-19

- Improved form element add drop-down order.
- Add initial migrations for DjangoCMS integration app.
- Add French translations.
- Implemented styles for `bleach` within `content_richtext` plugin.
- Documentation improvements.

48.2.30 0.12.17

2018-02-14

- Security fixes in `db_store` plugin.
- Minor fixes and cleanups.
- If CKEditor is installed, use it (rich text) for success page message.

48.2.31 0.12.16

2018-01-21

Note: Note, that this release contains minor backwards incompatible changes, that may slightly break your JS and/or styling.

The *form-horizontal* class attribute on the main form with elements (in templates) has been replaced with *fobi-form* id attribute.

Affected files:

- `src/fobi/templates/fobi/generic/add_form_element_entry_ajax.html`
- `src/fobi/templates/fobi/generic/edit_form_element_entry_ajax.html`
- `src/fobi/templates/fobi/generic/snippets/form_ajax.html`

If you have modified any of these, you might want to check and update your code.

- Invisible reCAPTCHA form plugin (security) added.

- Clean up templates.
- Introduce a `form_id` block. The following templates were affected.

48.2.32 0.12.15

2018-01-17

- Minor clean up of HTML templates (in some templates `hidden_form_element_wrapper_html_class` was used while others `form_element_wrapper_hidden_html_class`). Now `form_element_wrapper_hidden_html_class` is used everywhere. Some HTML files have been re-indented (2 spaces).
- Minor clean up of JS.
- Optionally limit file extensions for file upload element

48.2.33 0.12.14

2018-01-10

- Minor clean up of HTML templates (unused classes removed).
- Nicer captcha (simple captcha) for bootstrap3 and foundation5 themes.

48.2.34 0.12.13

2018-01-09

Note: Note, that this release contains minor backwards incompatible changes, that may slightly break your JS and/or styling.

The *form-horizontal* class attribute on the main form with elements (in templates) has been replaced with *fobi-form* id attribute.

Affected files:

- `/src/fobi/contrib/themes/djangocms_admin_style_theme/static/djangocms_admin_style_theme/js/fobi.djangocms_admin_style_th`
- `/src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html`
- `/src/fobi/contrib/themes/foundation5/static/foundation5/js/foundation5_fobi_extras.js`
- `/src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html`
- `/src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_wizard_entry_ajax.html`
- `/src/fobi/contrib/themes/simple/static/simple/js/fobi.simple.edit.js`
- `/src/fobi/contrib/themes/simple/templates/simple/edit_form_entry_ajax.html`
- `/src/fobi/contrib/themes/simple/templates/simple/edit_form_wizard_entry_ajax.html`
- `/src/fobi/contrib/themes/bootstrap3/static/bootstrap3/js/bootstrap3_fobi_extras.js`
- `/src/fobi/templates/fobi/generic/edit_form_entry_ajax.html`

If you have modified any of these, please double check that your edit views work and look properly.

- Fixes in simple and django-cms-admin-style themes (assets being loaded incorrectly since 0.12.6).

- Base template, as well as templates of all the themes have been made a little bit more flexible.
- Minor documentation fixes.

48.2.35 0.12.12

2018-01-03

- More consistent loading of assets (JS, CSS) in add- and edit-form-element-entry templates.
- Copyright and licenses year update.
- Minor fixes in the widgets.
- Clean up.

48.2.36 0.12.11

2017-12-27

- Clean up Django < 1.8 code.
- Clean up old code.

48.2.37 0.12.10

2017-12-24

- Django 2.0 support (experimental).
- (Temporary) replace `django-autoslug` package with `django-autoslug-iplweb`, which works fine for Django versions `>=1.8,<=2.0`.

48.2.38 0.12.9

2017-12-21

- Added content markdown plugin.

48.2.39 0.12.8

2017-12-19

- Added common form callback `AutoFormMail` for auto mailing the form data (without need to add the mail plugin to the form).

48.2.40 0.12.7

2017-12-17

- Minor fixes (in the add form element entry bootstrap3 template).
- Add content rich text plugin (based on CKEditor).
- Added common form callback `AutoFormDbStore` for auto saving the form data (without need to add the `db_store` plugin to the form).

48.2.41 0.12.6

2017-12-12

- Clean up.
- Minor documentation fixes.
- Minor plugin clean-up/fixes (`captcha`, `recaptcha`, `content_text`).
- Minor Python 3 fixes.
- Minor fixes in `FormWizard` `get_absolute_url` method.

48.2.42 0.12.5

2017-09-27

- Documentation fixes.

48.2.43 0.12.4

2017-09-27

- Documentation fixes.

48.2.44 0.12.3

2017-09-27

- Minor fixes.
- Reflect form-wizards support changes in the *foundation5* theme.
- Documentation fixes.

48.2.45 0.12.2

2017-08-02

- Some work on full form-wizards support in the *foundation5* theme.
- Django 1.11 fixes for customised widgets.
- Update example project requirements.

48.2.46 0.12.1

2017-07-31

- Reflect form-wizards support changes in the *simple* theme.
- Fixes in docs.

48.2.47 0.12

2017-06-28

- Wagtail integration (yet experimental).

48.2.48 0.11.13

2017-06-10

- Updated docs for DjangoCMS, FeinCMS and Mezzanine.
- Updated outdated Dutch translations.
- Improvements of the DRF integration app documentation.
- Duration field added.
- Support DurationField in `drf_integration` integration app.
- Minor fixes in docs.

48.2.49 0.11.12

2017-05-31

- Added a lot of field metadata to the `OPTIONS` call of `drf_integration` app.
- Appended a lot of sub-module README files to the main documentation.

48.2.50 0.11.11

2017-05-29

- Minor fixes in `content_text drf_integration` plugin.
- Added `imageurl` support to the `mailchimp_importer` plugin.

48.2.51 0.11.10

2017-05-26

- Minor fixes in form-wizards on Django 1.11.

48.2.52 0.11.9

2017-05-24

- Mezzanine integration updated to work with Mezzanine 4.2.3.
- Fixes in date-drop-down plugin when using form wizards.

48.2.53 0.11.8

2017-05-17

- `ContentImageURL` plugin added.
- Minor Python3 fixes in `db_store` plugin (related to export of forms to `xls` format).

48.2.54 0.11.7

2017-05-16

- Fixed in `fobi.integration` package related to Django 1.10 and higher.
- `FeinCMS` integration updated (only migrations added) to work with `FeinCMS` 1.13.2.
- `DjangoCMS` integration updated to work with `DjangoCMS` 3.4.3.

48.2.55 0.11.6

2017-05-15

- Minor fixes in `drf_integration` app, added GET/detail actions tests.

48.2.56 0.11.5

2017-05-15

- Added `date_drop_down` to `drf_integration` app.
- Fixed dependencies issue.
- Added dedicated requirements for specific Django versions.

48.2.57 0.11.4

2017-05-12

- Minor fixes in integration callbacks of the `drf_integration` sub-package.
- Added support for `content_image`, `content_text` and `content_video` plugins.
- Fixes in installable demo.

48.2.58 0.11.3

2017-05-10

- Concept of integration callbacks introduced and implemented for the `drf_integration` sub-package.

48.2.59 0.11.2

2017-05-09

- Minor fixes in `drf_integration` app.

48.2.60 0.11.1

2017-05-08

- Minor fixes in `decimal` plugin.
- Minor documentation improvements.

48.2.61 0.11

2017-05-07

This release is dedicated to my beloved [wife](#) for all the love and support she gave me through years. If you are a company looking for (female) developers in Groningen area (the Netherlands), do not hesitate to [contact her](#).

- Django REST framework integration. Check the Heroku demo [here](#).
- Documentation fixes.
- PEP8 code fixes.
- Minor setup fixes related to moved screen-shots file.
- Added helper scripts to test with Firefox in headless mode. Describe testing with Firefox in headless mode in documentation.

- Validate the decimal field plugin - quantize the decimal value to the configured precision.
- Minor fixes in the float field plugin.
- Minor improvements in complex form element plugins (select, file) and form handler plugins (db_store, mail, http_repost) in order to simplify integration plugins and reduce code duplication.
- Minor Python3 fixes in range_select and slider form element plugins.
- Minor Python3 fixes in http_repost and mail form handler plugins.

48.2.62 0.10.7

2017-03-13

- Several Django deprecation/moves fixes for better future compatibility.

48.2.63 0.10.6

2017-02-14

- Minor Python 3 fixes for integer, float and decimal fields.

48.2.64 0.10.5

2017-02-13

- Tested against Python 3.6.
- Initial (experimental) Django 1.11 support.

48.2.65 0.10.4

2017-01-11

- Minor fixes in Django admin.
- Various pep8 fixes.
- Fixes additions and improvements in/of docs.
- Add options to test with PhantomJS instead of Firefox.

48.2.66 0.10.3

2016-11-24

- Minor fixes.

48.2.67 0.10.2

2016-11-24

- Minor fixes.

48.2.68 0.10.1

2016-11-17

- Fixed captcha and re-captcha issues in form wizards.

48.2.69 0.10

2016-11-16

Note: Note, that this release contains minor backwards incompatible changes, that may break your code. Two additional arguments have been added to the *submit_plugin_form_data* method of the form element plugins. If you have written custom form element plugins - update your code.

- Added *form_entry_elements* and *kwargs* to the *submit_plugin_form_data* method of the form element plugins. Make sure to update your custom plugins if you have written any.
- Added tests for mailchimp integration plugin.
- Moving all plugins to base submodules of the correspondent sub packages.
- Add missing whitespace to the *help_text* of the *title* field of *FormEntry* and *FormWizardEntry* models.
- Disable GoogleAnalytics while testing (guess what - this change speeds up selenium tests twice).
- Docs updated.
- Helper scripts updated.
- Multiple pep8 fixes.

48.2.70 0.9.17

2016-11-13

Note: Announcing dropping support of Python 2.6 and Django 1.7. As of 0.9.17 everything is still backwards compatible with Django 1.7, but in future versions it will be wiped out.

- Value validations for Integer and Text Fields.
- Hide previous button in form wizard template for bootstrap3 on first step.

48.2.71 0.9.16

2016-11-10

- Introduced form titles (shown in view templates).
- Improved navigation of the form wizards.

48.2.72 0.9.15

2016-11-07

- Minor fixes.

48.2.73 0.9.14

2016-11-07

- Minor fixes.

48.2.74 0.9.13

2016-11-05

Note: Announcing dropping support of Django 1.5 and 1.6. As of 0.9.13 everything is still backwards compatible with versions 1.5 and 1.6, but in future versions compatibility with these versions will be wiped out.

- Fix backwards compatibility of *slider* and *range_select* plugins with Django versions 1.5 and 1.6.

48.2.75 0.9.12

2016-11-02

- Better debugging.
- Upgrade example FeinCMS integration to work with 1.12.

48.2.76 0.9.11

2016-11-01

- Fixes.

48.2.77 0.9.10

2016-11-01

- Fixed issue with custom labels in the *slider* plugin.
- Made *slider* plugin compatible with Django <= 1.6.
- Fixes *get_absolute_url* methods on *FormEntry* and *FormWizardEntry* models. #48

48.2.78 0.9.9

2016-10-31

- Make it possible to add custom ticks to the *slider* plugin.

48.2.79 0.9.8

2016-10-27

- Support multiple sliders in one form.

48.2.80 0.9.7

2016-10-27

- Improvements in the generic integration processor. #47
- Improved form wizard interface and navigation.
- Fixed a broken test.
- Added import/export functionality for form wizards.

48.2.81 0.9.6

2016-10-25

- Fixed *InvalidQuery* exception raised when attempting to export entry from a 'DB store' handler. #44
- Fixed *ProgrammingError* raised when using the 'Export data to CSV/XLS' action. #45

48.2.82 0.9.5

2016-10-25

- Minor fixes in *slider* and *select_range* plugins.

48.2.83 0.9.4

2016-10-24

- Fix issue with *select_multiple*, *select_multiple_model_objects* and *select_multiple_mptt_model_objects* being invalidated on the last step of the form wizard.

48.2.84 0.9.3

2016-10-24

- Change to *NumberInput* widget for all number inputs.
- Fixed issue with *slider* plugin missing labels if *Show endpoints as* is set to *Labeled ticks*.
- Link to edit form entry added to edit form wizard entry view.

48.2.85 0.9.2

2016-10-24

- Minor fixes.

48.2.86 0.9.1

2016-10-24

- Minor fixes.

48.2.87 0.9

2016-10-24

Note: Note, that this release contain minor backwards incompatible changes, that may break your existing code (your data is left intact). If you have written custom form element plugins you should update your code!

- The `:method:`get_form_field_instances`` and `:method:`_get_form_field_instances`` of the `fobi.base.FormElementPlugin` both accept two new optional arguments: `form_entry` and `form_element_entries` as well as `**kwargs`. Make sure to update your custom plugins if you have written any.
- Minor fixes in the form wizards: forms in intermediate steps do receive updates from the `submit_plugin_form_data` of the plugins.
- Fixed issue in the `base_bulk_change_plugins` function on Django 1.10.

48.2.88 0.8.10

2016-10-22

- Minor CSS improvements of the *slider* plugin.
- Fixed broken readthedocs requirements.

48.2.89 0.8.9

2016-10-22

- Simplified debugging (never set *FOBI_DEBUG* to True in production!).
- Major *slider* plugin improvements.

48.2.90 0.8.8

2016-10-21

- Minor *slider* plugin improvements (JavaScript).

48.2.91 0.8.7

2016-10-21

- Fixed issue of plugin media not being collected in the form wizard.

48.2.92 0.8.6

2016-10-21

- Functional improvements of *slider* plugin.

48.2.93 0.8.5

2016-10-20

- Add *range_select* and *slider* form field plugins.
- Fixed custom CSS classes not appearing in the rendered HTML of the field plugin/widget.
- Fixed issue with undefined file storage for form wizards. From now on the *FileSystemStorage* storage is used for wizard uploads.
- Fixed too much of extreme data view/export security of the *db_store* plugin.
- Backwards compatibility fixes for Django < 1.7.

48.2.94 0.8.4

2016-10-19

- Fix broken export (to JSON) of form entries.
- Fix broken import (from JSON) of form entries.

48.2.95 0.8.3

2016-10-18

- Minor fixes.

48.2.96 0.8.2

2016-10-18

- Minor fixes.

48.2.97 0.8.1

2016-10-17

- Minor fixes.

48.2.98 0.8

2016-10-17

Release supported by [Lund University Cognitive Science](#).

- Adding form-wizards functionality. Note, that at the moment only *bootstrap3* theme was updated to fully support the form wizards. Although, all other themes would by default support form-wizard functionality, they may not look as nice as they should be (to be fixed in 0.8.x releases shortly).
- The *six* package requirements increased to ≥ 1.8 .
- Tests comply with pep8.
- Fixed recently broken drag-and-drop ordering of the form elements.
- Fixed typo for HTML id “tab-form-elemenets” -> “tab-form-elements”. You may need to update your custom CSS/JS/HTML accordingly. See the listing 0.8.a for the files affected.
- An additional property *form_view_form_entry_option_class* has been added to all the themes. Change your custom CSS/JS/HTML accordingly. See the listing 0.8.b for the files affected.
- Fixed drag-and-drop not working for ordering of form elements. #43
- Fixed issue with non-proper rendering of the form-importer templates.

Note: Although this release does not contain backwards incompatible changes, there have been several changes in GUI and some parts of the generic HTML and themes were updated. If you have custom themes implemented, you should likely make some minor updates to the HTML in order to reflect the latest GUI changes. The following templates have been affected:

48.2.98.1 New files

- `src/fobi/contrib/plugins/form_handlers/db_store/templates/db_store/view_saved_form_wizard_data_entries.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/add_form_wizard_handler_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/add_form_wizard_handler_entry_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/create_form_wizard_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/create_form_wizard_entry_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/edit_form_wizard_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/edit_form_wizard_entry_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/form_wizards_dashboard.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/snippets/form_wizard_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/snippets/form_wizard_properties_snippet.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/snippets/form_wizard_snippet.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/view_form_wizard_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/view_form_wizard_entry_ajax.html`
- `src/fobi/templates/fobi/generic/add_form_wizard_handler_entry.html`
- `src/fobi/templates/fobi/generic/add_form_wizard_handler_entry_ajax.html`
- `src/fobi/templates/fobi/generic/create_form_wizard_entry.html`
- `src/fobi/templates/fobi/generic/create_form_wizard_entry_ajax.html`
- `src/fobi/templates/fobi/generic/edit_form_wizard_entry.html`
- `src/fobi/templates/fobi/generic/edit_form_wizard_entry_ajax.html`
- `src/fobi/templates/fobi/generic/form_wizard_entry_submitted.html`
- `src/fobi/templates/fobi/generic/form_wizard_entry_submitted_ajax.html`
- `src/fobi/templates/fobi/generic/form_wizards_dashboard.html`
- `src/fobi/templates/fobi/generic/snippets/form_wizard_ajax.html`
- `src/fobi/templates/fobi/generic/snippets/form_wizard_properties_snippet.html`
- `src/fobi/templates/fobi/generic/snippets/form_wizard_snippet.html`
- `src/fobi/templates/fobi/generic/snippets/form_wizard_view_ajax.html`
- `src/fobi/templates/fobi/generic/view_form_wizard_entry.html`
- `src/fobi/templates/fobi/generic/view_form_wizard_entry_ajax.html`

48.2.98.2 Existing files

- `src/fobi/contrib/plugins/form_importers/mailchimp_importer/templates/mailchimp_importer/1.html`
- `src/fobi/contrib/plugins/form_importers/mailchimp_importer/views.py`
- `src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html`
- `src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html`
- `src/fobi/templates/fobi/generic/edit_form_entry_ajax.html`

48.2.98.3 Additional listings

48.2.98.3.1 Listing 0.8.a “tab-form-elemenets” -> “tab-form-elements”

- `src/fobi/contrib/themes/djangocms_admin_style_theme/static/djangocms_admin_style_theme/css/fobi.djangocms_admin_style_th`
- `src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html`
- `src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html`
- `src/fobi/contrib/themes/simple/static/simple/css/fobi.simple.edit.css`
- `src/fobi/contrib/themes/simple/templates/simple/edit_form_entry_ajax.html`
- `src/fobi/templates/fobi/generic/edit_form_entry_ajax.html`
- `src/fobi/templates/fobi/generic/edit_form_wizard_entry_ajax.html`

48.2.98.3.2 Listing 0.8.b *form_view_form_entry_option_class* property

- `src/fobi/contrib/themes/bootstrap3/fobi_themes.py`
- `src/fobi/contrib/themes/djangocms_admin_style_theme/fobi_themes.py`
- `src/fobi/contrib/themes/foundation5/fobi_themes.py`
- `src/fobi/contrib/themes/simple/fobi_themes.py`
- `src/fobi/templates/fobi/generic/edit_form_wizard_entry_ajax.html`

48.2.99 0.7.1

2016-09-22

Release sponsored by [Goldmund](#), [Wyldebeast](#) & [Wunderliebe](#).

- Code comply with pep8.
- Minor fixes in selenium tests.

48.2.100 0.7

2016-09-13

Release sponsored by [Goldmund, Wyldebeast & Wunderliebe](#).

- Initial Django 1.10 support.
- *django-localeurl* has been replaced with *i18n_patterns* in sample project.
- Minor fixes.

48.2.101 0.6.10

2016-09-11

- Moved *plugin_uid* field choices from model level to form level for *FormHandler* and *FormHandlerEntry* models. #37

48.2.102 0.6.9

2016-09-08

- Moved *plugin_uid* field choices from model level to form level for *FormElement* and *FormElementEntry* models. #37
- Fixed element “name” field stripping underscores issue. #33

48.2.103 0.6.8

2016-09-06

- Fixed changing order of the *FormElement*, *FormElementEntry*, *FormHandler* and *FormHandlerEntry* models.

48.2.104 0.6.7

2016-08-30

- Minor fixes in *db_store* plugin (Django 1.6 compatibility issue).
- Added `__str__` methods to models.
- Restrict queryset to form element entries related to the form entry in position calculation.

48.2.105 0.6.6

2016-05-12

- Fixed broken dependencies in demos.
- Minor fixes.
- Adding a new *Select multiple with max* plugin, which restricts max number of choices allowed to be chosen.

48.2.106 0.6.5

2015-12-24

- Minor fixes.

48.2.107 0.6.4

2015-12-24

- Minor fixes.

48.2.108 0.6.3

2015-12-23

- Minor fixes.

48.2.109 0.6.2

2015-12-22

- Make it possible to render a list of forms using custom template tag (not only on the dashboard page).

48.2.110 0.6.1

2015-12-21

- Documentation fixes.

48.2.111 0.6

2015-12-18

- Form importers (and as a part of it - MailChimp integration, which would allow to import forms from MailChimp into *django-fobi* using a user-friendly wizard).
- Improved Django 1.9 support.

48.2.112 0.5.19

2015-12-15

- New style urls everywhere.

48.2.113 0.5.18

2015-12-08

- Minor improvements. Adding request to the *get_form_field_instances* method of the *FormElementPlugin*.

48.2.114 0.5.17

2015-10-22

- Increased *easy-thumbnails* requirement to ≥ 2.1 for Python 3, since it was causing installation errors.
- Increased the *django-nine* requirement to $\geq 0.1.6$, since it has better Python 3 support.

48.2.115 0.5.16

2015-10-05

- Fixes in migrations on Django 1.7.

48.2.116 0.5.15

2015-09-17

- Minor fixes.

48.2.117 0.5.14

2015-09-13

- Add create/update dates to the forms. Adding initial migrations.

48.2.118 0.5.13

2015-09-01

- Translation fixes.
- Updating Dutch and Russian translations.

48.2.119 0.5.12

2015-08-29

- Export/import forms into/from JSON.
- Minor UI improvements. Adding “Service” tab in the edit view which contains links to export form to JSON and delete form.

Note: Although this release does not contain backwards incompatible changes, there have been several changes in GUI and some parts of the generic HTML and themes were updated. If you have custom themes implemented, you should likely make some minor updates to the HTML in order to reflect the latest GUI changes. The following templates have been affected:

48.2.119.1 New files

- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/import_form_entry.html
- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/import_form_entry_ajax.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/import_form_entry.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/import_form_entry_ajax.html
- src/fobi/contrib/themes/simple/templates/simple/import_form_entry.html
- src/fobi/contrib/themes/simple/templates/simple/import_form_entry_ajax.html
- src/fobi/templates/fobi/generic/import_form_entry.html
- src/fobi/templates/fobi/generic/import_form_entry_ajax.html

48.2.119.2 Existing files

- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/dashboard.html
- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/dashboard.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html
- src/fobi/contrib/themes/simple/templates/simple/dashboard.html
- src/fobi/contrib/themes/simple/templates/simple/edit_form_entry_ajax.html
- src/fobi/templates/fobi/generic/dashboard.html
- src/fobi/templates/fobi/generic/edit_form_entry_ajax.html

48.2.120 0.5.11

2015-08-20

- Minor improvements of the dynamic values feature. Forbid usage of django template tags in initial values.

48.2.121 0.5.10

2015-08-18

- Minor improvements of the initial dynamic values feature.

48.2.122 0.5.9

2015-08-17

- Minor fixes in the initial dynamic values feature.

48.2.123 0.5.8

2015-08-16

- Made it possible to define dynamic initials for form fields. Example initial dynamic values in the form (like `{{ request.path }}`).
- Minor fixes/improvements.

48.2.124 0.5.7

2015-08-03

- Minor Python 3 improvements.

48.2.125 0.5.6

2015-07-31

- *django-mptt* support through *select_mptt_model_object* and *select_multiple_mptt_model_objects* plugins.
- Python 3 fixes.

48.2.126 0.5.5

2015-06-30

- Change the *action* field of the *FormEntry* into a URL field; check if action exists.
- *captcha*, *recaptcha* and *honeypot* plugins have been made required in the form.
- Fix: take default values provided in the *plugin_data_fields* of the plugin form into consideration.

48.2.127 0.5.4

2015-05-21

- Minor Django 1.8 fixes.
- Improved texts/translations.

48.2.128 0.5.3

2015-05-02

- Minor fixes in the [mail](#) form handler plugin.

48.2.129 0.5.2

2015-04-26

- Make it possible to provide multiple *to* email addresses in the `mail` form handler plugin.
- DateTime picker widget added for Foundation5 theme for `date` and `datetime` plugins.
- Added more tests (more plugins tested).

48.2.130 0.5.1

2015-04-21

- DateTime picker widget added for Bootstrap 3 theme for `date` and `datetime` plugins.

48.2.131 0.5

2015-04-06

Note: Note, that this release contains minor backwards incompatible changes. The changes may affect your existing forms and data. Read the notes below carefully.

- Fixed previously wrongly labeled (in *AppConf*) add-ons/plugins (*fobi.contrib.plugins.form_handlers.db_store*, *fobi.contrib.apps.feincms_integration*, *fobi.contrib.apps.djangocms_integration*, *fobi.contrib.apps.mezzanine_integration*). Due to the change, you would likely have to rename a couple of database tables and update references accordingly. No migrations to solve the issue are included at the moment.

48.2.132 0.4.36

2015-04-03

- Handle non-ASCII characters `content_text` form element.

48.2.133 0.4.35

2015-03-28

- Fixed the issue with *allow_multiple* working incorrectly for form handler plugins. Fix the *db_store* plugin as well.

48.2.134 0.4.34

2015-03-27

- Minor fixes in the *Checkbox select multiple* and *Radio* plugins.
- Minified tox tests.

48.2.135 0.4.33

2015-03-26

- `Checkbox select multiple` field added.
- Minor improvements (styling) in the Foundation 5 theme.
- Initial configuration for tox tests.
- Clean up requirements (for example setups and tests).

48.2.136 0.4.32

2015-03-25

- Updated missing parts in the Russian translations.
- Minor API improvements. From now on, the `run` method of form handlers may return a tuple (bool, mixed). In case of errors it might be (False, err).
- Minor code clean ups.

48.2.137 0.4.31

2015-03-23

- When path of the uploaded file (plugins) doesn't yet exist, create it, instead of failing.

48.2.138 0.4.30

2015-03-23

- From now on submitted files are sent as attachments in the mail plugin.
- Documentation improvements. Adding information of rendering forms using *django-crispy-forms* or alternatives.
- Minor fixes.

48.2.139 0.4.29

2015-03-20

- `Decimal` field added.
- `Float` field added.
- `Slug` field added.
- `IP address` field added.
- `Null boolean` field added.
- `Time` field added.
- From now on using *simplejson* package in favour of *json*, since it can handle decimal data.
- Minor improvements of the date plugins (*datetime*, *date*).

48.2.140 0.4.28

2015-03-13

- Fix improperly picked configurations of the *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects* plugin.
- Long identifiers of models can now be safely used in foreign key plugins (such as *fobi.contrib.plugins.form_elements.fields.select_model_object_plugin* and *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects* plugins).
- Fixed admin bulk change of the plugins.
- From now it's possible to have some control/configure the following plugins for the submitted value:
 - *fobi.contrib.plugins.form_elements.fields.select_model_object*
 - *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects*

The only thing needs to be done is to specify the appropriate variable in the settings module of the project (settings.py).

- `FOBI_FORM_ELEMENT_SELECT_MODEL_OBJECT_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MODEL_OBJECTS_SUBMIT_VALUE_AS`

Allowed values are: “val”, “repr”, “mix”.

48.2.141 0.4.27

2015-03-12

- Temporary allow the *fobi.contrib.plugins.form_handlers.db_store* plugin to be used multiple times per form, until the bug with not being able to assign the *db_store* plugin to the form due to incorrect handling of restrictions (`allow_multiple`) introduced in previous version is properly fixed.
- From now it's possible to have some control/configure the following plugins for the submitted value:
 - *fobi.contrib.plugins.form_elements.fields.radio*
 - *fobi.contrib.plugins.form_elements.fields.select*
 - *fobi.contrib.plugins.form_elements.fields.select_multiple*

The only thing needs to be done is to specify the appropriate variable in the settings module of the project (settings.py).

- `FOBI_FORM_ELEMENT_RADIO_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_SUBMIT_VALUE_AS`

Allowed values are: “val”, “repr”, “mix”.

48.2.142 0.4.26

2015-03-06

- Validate *fobi.contrib.plugins.form_elements.fields.email*, *fobi.contrib.plugins.form_elements.fields.integer* and *fobi.contrib.plugins.form_elements.fields.url* plugins initial values.
- Properly show field types “checkbox” and “radio” in the *input* plugin (as previously they showed up too large).
- It’s now possible to restrict multiple usage of form handler plugins via *allow_multiple* property. In case if it’s set to False, the plugin can be used once only (per form). Default value is True. The *db_store* plugin is from now on allowed to be used only once (per form).

48.2.143 0.4.25

2015-03-04

- Post-fix in the discover module (moved logging definition up).

48.2.144 0.4.24

2015-03-04

- The management command *migrate_03_to_04* intended to migrate 0.3.x branch data to > 0.4.x branch data, has been renamed to *fobi_migrate_03_to_04*.
- Add missing app config declaration for the *db_store* form handler plugin.
- Add missing app config for the core *fobi* package.
- Improved autodiscover for Django>=1.7. Fix exception when using a dotted path to an *AppConfig* in *INSTALLED_APPS* (instead of using the path to the app: ex. “path.to.app.apps.AppConfig” instead of “path.to.app”).

48.2.145 0.4.23

2015-03-04

- Fix improper initial value validation for Select-like (*radio*, *select* and *select_multiple*) plugins.

48.2.146 0.4.22

2015-03-03

- Fix replace system-specific path separator by a slash on file urls.
- Fix empty options appearing in the Select-like (*radio*, *select* and *select_multiple*) plugins and unified the processing of the raw choices data.
- Validate the initial value for Select-like (*radio*, *select* and *select_multiple*) plugins.

48.2.147 0.4.21

2015-02-28

- The `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects` plugin added.

48.2.148 0.4.20

2015-02-27

- Make it possible to load initial form data from GET variables.
- Remove “button” and “submit” types from `fobi.contrib.elements.fields.input` form element plugin.
- The `fobi.contrib.plugins.form_elements.fields.select_model_object` plugin no longer returns an absolute URL to the admin of the chosen model object instance. Instead, it contains the app label, model name, pk and the repr of it separated by dots. Example: “comments.comment.5.Lorem ipsum”.
- Minor fixes in from `fobi.contrib.elements.fields.file` plugin. Use system path separator on from `fobi.contrib.elements.fields.file.FILES_UPLOAD_DIR` default setting.
- Minor documentation improvements.

48.2.149 0.4.19

2015-02-15

- Some more work on future (Django 1.9) compatibility.
- Replace bits of backwards-/forwards- compatibility code with equal code parts of `django-nine`.

48.2.150 0.4.18

2015-02-13

- From now on it’s possible to localise (translated) URLs.
- Safe way to get the user model for Django `>= 1.5.* <= 1.8.*`

48.2.151 0.4.17

2015-02-12

- Fix circular imports by proper referencing of the user model in foreign key and many-to-many relations (`settings.AUTH_USER_MODEL` instead of `django.contrib.auth.get_user_model`).
- Minor documentation improvements.

48.2.152 0.4.16

2015-02-10

- A new theme `djangoCMS_admin_style_theme` added.
- Making `fobi.fields.NoneField` always valid.
- Initial work on Django 1.8 and 1.9 support.
- Minor fixes.

48.2.153 0.4.15

2015-01-27

- Fix the “large” checkboxes in edit mode - now shown small as they should be.

48.2.154 0.4.14

2015-01-26

- German translations added.

48.2.155 0.4.13

2015-01-15

- Remove an *ipdb* statement from base integration processor *fobi.integration.processors.IntegrationProcessor*.
- Added information in the docs about FeinCMS demo part on heroku demo.
- Make sure values of form elements declared not to have a value (`has_value` property is set to `False`) aren't being saved in the `db_store` plugin.
- Remove redundant static assets (package size decreased).

48.2.156 0.4.12

2015-01-14

- Fix empty options appearing in the Select-like plugins and unified the processing of the raw choices data.
- Update the *vishap* package requirement to latest stable 0.1.3.
- Support for wheel packages.

48.2.157 0.4.11

2014-12-29

- Styling fixes in the `radio` button field of the `bootstrap3` theme.
- Fixed `db_store` issue with CSV/XLS export failing on Django 1.7.

48.2.158 0.4.10

2014-12-28

- Minor fixes in FeinCMS integration app.

48.2.159 0.4.9

2014-12-28

- Third party app integration (at the moment, FeinCMS, DjangoCMS, Mezzanine) had been generalised and unified.
- Mention the Heroku live demo in the docs.
- Minor CSS fixes in the `simple` theme.

48.2.160 0.4.8

2014-12-25

- More verbose debugging.

48.2.161 0.4.7

2014-12-24

- Temporary left out the “cloneable” column from the dashboard templates.
- Fixed broken imports in CAPTCHA plugin.
- Fixed broken imports in ReCAPTCHA plugin.

48.2.162 0.4.6

2014-12-23

- Updated requirements for the `vishap` package to avoid the `six` version conflicts.
- Minor documentation fixes.

48.2.163 0.4.5

2014-12-17

- ReCAPTCHA field added.
- Mezzanine integration app added.
- Remove redundant dependencies (django-tinymce).
- Minor improvements of the discover module.

48.2.164 0.4.4

2014-12-06

- Documentation improvements.
- Updated Dutch and Russian translations.
- Minor fixes related to lazy translations.

48.2.165 0.4.3

2014-12-05

- Make sure values of form elements declared not to have a value (`has_value` property is set to `False`) aren't being saved in the `db_store` plugin.
- Apply that to the `honeypot` and `captcha` plugins.

48.2.166 0.4.2

2014-12-04

- Helper script (management command) in order to migrate `django-fobi==0.3.*` data to `django-fobi==0.4.*` data (caused by renaming the `birthday` field to `date_drop_down` - see the release notes of 0.4 below). Follow the steps precisely in order to painlessly upgrade your `django-fobi==0.3.*` to `django-fobi==0.4.*`:

1. Install `django-fobi>=0.4.2`:

```
pip install django-fobi>=0.4.2
```

2. In your settings change the:

```
'fobi.contrib.plugins.form_elements.fields.birthday'
```

to:

```
'fobi.contrib.plugins.form_elements.fields.date_drop_down'
```

3. Run the `migrate_03_to_04` management command. Note, that as of version 0.4.24, the `migrate_03_to_04` command has been renamed to `fobi_migrate_03_to_04`:

```
./manage.py migrate_03_to_04
```

48.2.167 0.4.1

2014-12-04

- Fixes in Foundation5 and Simple themes related to the changes in error validation/handling of hidden fields.

48.2.168 0.4

2014-12-03

Note: Note, that this release contains minor backwards incompatible changes. The changes may affect your existing forms and data. Read the notes below carefully (UPDATE 2014-12-04: the django-fobi==0.4.2 contains a management command which makes the necessary changes in the database for safe upgrade).

- The captcha field has been moved from `fobi.contrib.plugins.form_elements.fields.captcha` to `fobi.contrib.plugins.form_elements.security.captcha`. Make sure to update the package paths in `INSTALLED_APPS` of your projects' settings module (`settings.py`) when upgrading to this version.
- The honeypot field has been added.
- The birthday field has been renamed to `date_drop_down` (A real birthday field is still to come in later releases). The change causes backwards incompatibility issues if you have used that birthday field. If you haven't - you have nothing to worry. If you have been using it, grab the 0.3.4 version, copy the `fobi.contrib.plugins.form_elements.fields.date_drop_down` package to your project apps, make necessary path changes and update the package paths in `INSTALLED_APPS` settings module (`settings.py`) before upgrading to this version. Then, in Django admin management interface, replace all the occurrences of Birthday field with Date drop down field.
- Better error validation/handling of hidden fields. A new form snippet template added for displaying the non-field and hidden fields errors. The new template makes a part of a standard theme as an attribute `form_non_field_and_hidden_errors_snippet_template`.
- Minor fixes in generic templates.
- An additional property `is_hidden` added to the hidden form elements. Those form elements would be getting a default TextInput widget in the edit mode instead of the widget they come from by default. It's possible to provide an alternative widget for the edit mode as well. Default value of the `is_hidden` is set to False.

48.2.169 0.3.4

2014-11-23

- New settings `FOBI_FAIL_ON_ERRORS_IN_FORM_ELEMENT_PLUGINS` and `FOBI_FAIL_ON_ERRORS_IN_FORM_HANDLER_PLUGIN` introduced. They do as their name tells. Default value for both is False.
- Fixed exceptions raised when unicode characters were used as form names.
- Fixed exceptions raised when unicode characters were used as field labels.
- Fixes in the `db_store` and `mail` plugins related to usage of unicode characters.

48.2.170 0.3.3

2014-11-22

- Clean up the setup. Remove redundant dependencies.
- Documentation improvements.

48.2.171 0.3.2

2014-11-20

- DjangoCMS integration app made compatible with DjangoCMS 2.4.3.

48.2.172 0.3.1

2014-11-19

- DjangoCMS integration app.

48.2.173 0.3

2014-11-09

Note: Note, that this release contains minor backwards incompatible changes. The changes do not anyhow affect your existing forms or data. The only thing you need to do is update the app paths in the `settings` module of your project.

- Minor core improvements related to the theming of the form handler plugins.
- Several presentational form element plugins have been renamed. The `fobi.contrib.plugins.form_elements.content.image` plugin has been renamed to `fobi.contrib.plugins.form_elements.content.content_image`. The `fobi.contrib.plugins.form_elements.content.text` plugin has been renamed to `fobi.contrib.plugins.form_elements.content.content_text`. The `fobi.contrib.plugins.form_elements.content.video` plugin has been renamed to `fobi.contrib.plugins.form_elements.content.content_video`. If you have used any of the above mentioned plugins, make sure to update the app paths in the `settings` module of your project.
- The `fobi.contrib.plugins.form_elements.content.dummy` plugin has been moved to `fobi.contrib.plugins.form_elements.test.dummy` location. If you have used it, make sure to update the its' path in the `settings` module of your project.
- Added readme to the following content form element plugins: `dummy`, `content_image`, `content_text` and `content_video`.
- Added `foundation5` and `simple` theme widgets for `db_store` plugin.
- If you have been overriding the defaults of the `db_store` plugin, change the prefix from `FOBI_PLUGIN_DB_EXPORT_` to `FOBI_PLUGIN_DB_STORE_`. For example, `FOBI_PLUGIN_DB_EXPORT_CSV_DELIMITER` should become `FOBI_PLUGIN_DB_STORE_CSV_DELIMITER`.
- Mentioning the `fobi.find_broken_entries` management command in the documentation, as well as improving the management command itself (more verbose output).
- Birthday field added.

48.2.174 0.2.1

2014-11-06

- Minor improvements of the `db_store` plugin.
- Minor improvements of the `simple` theme. Make sure that custom form handler actions are properly shown in the form handlers list.
- Make it possible to fail silently on missing form element or form handler plugins by setting the respected values to `False`: `FOBI_FAIL_ON_MISSING_FORM_ELEMENT_PLUGINS`, `FOBI_FAIL_ON_MISSING_FORM_HANDLER_PLUGINS`. Otherwise an appropriate exception is raised.

48.2.175 0.2

2014-11-05

Note, that this release contains minor backwards incompatible changes.

- Minor (backwards incompatible) changes in the form handler plugin API. From now on both `custom_actions` and `get_custom_actions` methods accept `form_entry` (obligatory) and `request` (optional) arguments. If you have written your own or have changed existing form handler plugins with use of one of the above mentioned methods, append those arguments to the method declarations when upgrading to this version. If you haven't written your own or changed existing form handler plugins, you may just upgrade to this version.
- Added data export features to the `db_store` plugin.
- Minor fixes in `db_store` plugin.
- Added missing documentation for the `feincms_integration` app.
- Updated translations for Dutch and Russian.

48.2.176 0.1.6

2014-10-25

- Minor improvements in the theming API. From now on the `view_embed_form_entry_ajax_template` template would be used when integrating the form rendering from other products (for example, a CMS page, which has a widget which references the form object. If that property is left empty, the `view_form_entry_ajax_template` is used. For a success page the `embed_form_entry_submitted_ajax_template` template would be used.
- Functional improvements of the FeinCMS integration (the widget). If you have used the FeinCMS widget of earlier versions, you likely want to update to this one. From now on you can select a custom form title and the button text, as well as provide custom success page title and the success message; additionally, it has been made possible to hide the form- or success-page- titles.

48.2.177 0.1.5

2014-10-23

- Minor fixes in the Integer form element plugin.
- Minor fixes in the Input form element plugin.
- Minor fixes in themes (disable HTML5 form validation in edit mode).
- Minor documentation improvements.

48.2.178 0.1.4

2014-10-22

- Minor core improvements.
- Django 1.5 support improvements.
- Django 1.7 support improvements.
- Added Captcha form element plugin.
- Added highly-customisable Input form element plugin - a custom input field with support for almost any ever existing HTML attribute.
- Documentation improvements.

48.2.179 0.1.3

2014-10-13

- Django 1.7 support.
- Add HTML5 “placeholder” field attribute support.

48.2.180 0.1.2

2014-10-11

- Simple theme fixes: Fix for making the theme work in Django 1.5.

48.2.181 0.1.1

2014-10-11

- Bootstrap 3 theme fixes: When tab pane has no or little content so that the height of the dropdown menu exceeds the height of the tab pane content the dropdown menu now becomes scrollable (vertically).

48.2.182 0.1

2014-10-11

- Initial release.

48.3 Licenses

Below information about third-party packages used in the project is presented.

48.3.1 Bootstrap3

<http://getbootstrap.com/>

Licensing information:

- Apache license 2.0 (<http://www.apache.org/licenses/LICENSE-2.0>)

48.3.2 Bootstrap 3 DateTime picker

<http://eonasdan.github.io/bootstrap-datetimepicker/>

- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.3 bootstrap-slider

<https://github.com/seiyria/bootstrap-slider>

- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.4 django-autoslug

<https://bitbucket.org/neithere/django-autoslug>

Licensing information:

- LGPL 3 (<https://www.gnu.org/licenses/lgpl.html>)

48.3.5 django-formtools

<https://github.com/django/django-formtools>

Licensing information:

- BSD (<http://opensource.org/licenses/BSD-3-Clause>)

48.3.6 django-nine

<https://github.com/barseghyanartur/django-nine>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)
- LGPL 2.1 (<https://www.gnu.org/licenses/lgpl-2.1.html>)

48.3.7 django-nonefield

<https://github.com/barseghyanartur/django-nonefield>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)
- LGPL 2.1 (<https://www.gnu.org/licenses/lgpl-2.1.html>)

48.3.8 easy-thumbnails

<https://github.com/SmileyChris/easy-thumbnails>

Licensing information:

- BSD (<http://opensource.org/licenses/BSD-3-Clause>)

48.3.9 Font-awesome

<http://fontawesome.io/>

Licensing information:

- <http://fontawesome.io/license/>
- SIL OFL 1.1 (<http://scripts.sil.org/OFL>)
- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.10 Foundation 5

<http://foundation.zurb.com/>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.11 Foundation 5 DateTime picker

<https://github.com/aliibrahim/foundation-datetimepicker-rails>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.12 jQuery Colorbox

<http://www.jacklmoore.com/colorbox/>

Licensing information:

- MIT (<http://www.opensource.org/licenses/mit-license.php>)

48.3.13 jQuery

<https://jquery.org>

Licensing information:

- <https://jquery.org/license/>
- MIT (<http://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt>)

48.3.14 jQueryUI

<http://jqueryui.com/>

Licensing information:

- MIT (<https://github.com/jquery/jquery-ui/blob/master/MIT-LICENSE.txt>)

48.3.15 jQuery-Slugify-Plugin

<https://github.com/pmcclhaney/jQuery-Slugify-Plugin>

Licensing information:

- BSD (<https://github.com/pmcclhaney/jQuery-Slugify-Plugin/blob/master/LICENSE>).

48.3.16 moment.js

<http://momentjs.com/>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.17 ordereddict

<https://pypi.python.org/pypi/ordereddict>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.18 Pillow

<http://python-pillow.github.io/>

Licensing information:

- Python Imaging Library license (<http://www.pythonware.com/products/pil/>)

48.3.19 Requests

<https://python-requests.org>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

48.3.20 Six

<https://bitbucket.org/gutworth/six/>

Licensing information:

- Apache 2.0 (<http://opensource.org/licenses/Apache-2.0>)

48.3.21 Unidecode

<https://pypi.python.org/pypi/Unidecode>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)

48.3.22 vishap

<https://pypi.python.org/pypi/vishap>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)
- LGPL 2.1 (<https://www.gnu.org/licenses/lgpl-2.1.html>)

48.4 fobi package

48.4.1 Subpackages

48.4.1.1 fobi.contrib package

48.4.1.1.1 Subpackages

48.4.1.1.1.1 fobi.contrib.apps package

48.4.1.1.1.2 Subpackages

48.4.1.1.1.3 fobi.contrib.apps.djangocms_integration package

48.4.1.1.1.4 Submodules

48.4.1.1.1.5 fobi.contrib.apps.djangocms_integration.apps module

```
class fobi.contrib.apps.djangocms_integration.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps.djangocms_integration'
    name = 'fobi.contrib.apps.djangocms_integration'
```

48.4.1.1.1.6 fobi.contrib.apps.djangocms_integration.cms_plugins module

48.4.1.1.1.7 fobi.contrib.apps.djangocms_integration.cms_version module

48.4.1.1.1.8 fobi.contrib.apps.djangocms_integration.conf module

```
fobi.contrib.apps.djangocms_integration.conf.get_setting(setting, override=None)
    Get setting.
```

Get a setting from `fobi.contrib.apps.djangocms_integration.conf` module, falling back to the default.

If `override` is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

48.4.1.1.1.9 fobi.contrib.apps.djangocms_integration.defaults module

48.4.1.1.1.10 fobi.contrib.apps.djangocms_integration.helpers module

`fobi.contrib.apps.djangocms_integration.helpers.get_form_template_choices()`

Get the form template choices.

It's possible to provide theme templates per theme or just per project.

Return list

`fobi.contrib.apps.djangocms_integration.helpers.get_success_page_template_choices()`

Get success page template choices.

Return list

48.4.1.1.1.11 fobi.contrib.apps.djangocms_integration.models module

48.4.1.1.1.12 fobi.contrib.apps.djangocms_integration.settings module

- `WIDGET_FORM_SENT` (str): Name of the GET param indicating that form has been successfully sent.

48.4.1.1.1.13 Module contents

48.4.1.1.1.14 fobi.contrib.apps.drf_integration package

48.4.1.1.1.15 Subpackages

48.4.1.1.1.16 fobi.contrib.apps.drf_integration.form_elements package

48.4.1.1.1.17 Subpackages

48.4.1.1.1.18 fobi.contrib.apps.drf_integration.form_elements.content package

48.4.1.1.1.19 Subpackages

48.4.1.1.1.20 fobi.contrib.apps.drf_integration.form_elements.content.content_image package

48.4.1.1.1.21 Submodules

48.4.1.1.1.22 fobi.contrib.apps.drf_integration.form_elements.content.content_image.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image.apps.Config(app_name,  
                                                                                       app_module)  
    Bases: django.apps.config.AppConfig  
    Config.  
    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_image'  
    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_image'
```


48.4.1.1.1.23 fobi.contrib.apps.drf_integration.form_elements.content.content_image.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image.base.ContentImagePlugin(
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSubmitPluginFormDataMixin

    CharField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content image'
    uid = 'content_image'
```

48.4.1.1.1.24 fobi.contrib.apps.drf_integration.form_elements.content.content_image.fobi_integration_form_element module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image.fobi_integration_form_element(
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSubmitPluginFormDataMixin

    CharField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content image'
    uid = 'content_image'
```

48.4.1.1.1.25 Module contents

48.4.1.1.1.26 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url package

48.4.1.1.1.27 Submodules

48.4.1.1.1.28 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.apps.Config(
    Bases: django.apps.config.AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_image_url'
```

```
name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_image_url'
```

48.4.1.1.1.29 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.base.ContentImageURLPlugin
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content image URL'
    uid = 'content_image_url'
```

48.4.1.1.1.30 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.fobi_integration_form_e module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content image URL'
    uid = 'content_image_url'
```

48.4.1.1.1.31 Module contents

48.4.1.1.1.32 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown package

48.4.1.1.1.33 Submodules

48.4.1.1.1.34 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.apps.Config(app_name,
                                                                                             app_module)
    Bases: django.apps.config.AppConfig
```

Config.

```
label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_markdown'
name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_markdown'
```

48.4.1.1.1.35 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.base.ContentMarkdownPlugin(
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin)
```

Content markdown (CharField) plugin.

```
get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                           form_element_entries=None, has_value=None, **kwargs)
```

Get form field instances.

```
group = 'Content'
```

```
integrate_with = 'drf_integration'
```

```
name = 'Content markdown'
```

```
uid = 'content_markdown'
```

48.4.1.1.1.36 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.fobi_integration_form_el module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.fobi_integration_form_el(
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin)
```

Content markdown (CharField) plugin.

```
get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                           form_element_entries=None, has_value=None, **kwargs)
```

Get form field instances.

```
group = 'Content'
```

```
integrate_with = 'drf_integration'
```

```
name = 'Content markdown'
```

```
uid = 'content_markdown'
```

48.4.1.1.1.37 Module contents

48.4.1.1.1.38 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext package

48.4.1.1.1.39 Submodules

48.4.1.1.1.40 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.apps.Config(app_name,
                                                                                          app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_richtext'
    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_richtext'
```

48.4.1.1.1.41 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base.ContentRichTextPlugin
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    Content rich text (CharField) plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content rich text'
    uid = 'content_richtext'
```

48.4.1.1.1.42 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.fobi_integration_form_el
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    Content rich text (CharField) plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content rich text'
```

```
uid = 'content_richtext'
```

48.4.1.1.1.43 Module contents

48.4.1.1.1.44 fobi.contrib.apps.drf_integration.form_elements.content.content_text package

48.4.1.1.1.45 Submodules

48.4.1.1.1.46 fobi.contrib.apps.drf_integration.form_elements.content.content_text.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_text.apps.Config(app_name,
                                                                                       app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_text'
    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_text'
```

48.4.1.1.1.47 fobi.contrib.apps.drf_integration.form_elements.content.content_text.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_text.base.ContentTextPlugin(user=None)
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content text'
    uid = 'content_text'
```

48.4.1.1.1.48 fobi.contrib.apps.drf_integration.form_elements.content.content_text.fobi_integration_form_element module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_text.fobi_integration_form_element
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
    integrate_with = 'drf_integration'
```

```
name = 'Content text'
uid = 'content_text'
```

48.4.1.1.1.49 Module contents

48.4.1.1.1.50 fobi.contrib.apps.drf_integration.form_elements.content.content_video package

48.4.1.1.1.51 Submodules

48.4.1.1.1.52 fobi.contrib.apps.drf_integration.form_elements.content.content_video.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_video.apps.Config(app_name,
                                                                                       app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_video'
    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_video'
```

48.4.1.1.1.53 fobi.contrib.apps.drf_integration.form_elements.content.content_video.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_video.base.ContentVideoPlugin(user)
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
    integrate_with = 'drf_integration'
    name = 'Content image'
    uid = 'content_video'
```

48.4.1.1.1.54 fobi.contrib.apps.drf_integration.form_elements.content.content_video.fobi_integration_form_element module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_video.fobi_integration_form_element
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = 'Content'
```

```
integrate_with = 'drf_integration'
name = 'Content image'
uid = 'content_video'
```

48.4.1.1.155 Module contents

48.4.1.1.156 Module contents

48.4.1.1.157 fobi.contrib.apps.drf_integration.form_elements.fields package

48.4.1.1.158 Subpackages

48.4.1.1.159 fobi.contrib.apps.drf_integration.form_elements.fields.boolean package

48.4.1.1.160 Submodules

48.4.1.1.161 fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps.Config(app_name,
                                                                                   app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_boolean'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.boolean'
```

48.4.1.1.162 fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base.BooleanSelectPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    Boolean select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Boolean'
    uid = 'boolean'
```

48.4.1.1.1.63 fobi.contrib.apps.drf_integration.form_elements.fields.boolean.fobi_integration_form_elements module

class fobi.contrib.apps.drf_integration.form_elements.fields.boolean.fobi_integration_form_elements.**Boolean**

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

Boolean select plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Boolean'

uid = 'boolean'

48.4.1.1.1.64 Module contents

48.4.1.1.1.65 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple package

48.4.1.1.1.66 Submodules

48.4.1.1.1.67 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.apps.**Config**(*app_name*, *app_model*)

Bases: [django.apps.config AppConfig](#)

Config.

label =

'fobi_contrib_apps_drf_integration_form_elements_fields_checkbox_select_multiple'

name =

'fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple'

48.4.1.1.1.68 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.base.**CheckboxSelectMultiple**

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

MultipleChoiceField plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

group = 'Fields'


```
integrate_with = 'drf_integration'
```

```
name = 'Checkbox select multiple'
```

```
submit_plugin_form_data(form_element_plugin, form_entry, request, serializer,
                        form_element_entries=None, **kwargs)
```

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** ([fobi.models.FormEntry](#)) – Instance of [fobi.models.FormEntry](#).
- **request** ([django.http.HttpRequest](#)) –
- **serializer** ([rest_framework.serializers.Serializer](#)) –
- **form_element_entries** ([iterable](#)) –

```
uid = 'checkbox_select_multiple'
```

48.4.1.1.169 [fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.fobi_integration_](#) module

```
class fobi.contrib.apps.drf\_integration.form\_elements.fields.checkbox\_select\_multiple.fobi\_integration\_
```

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

MultipleChoiceField plugin.

```
get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                           form_element_entries=None, **kwargs)
```

Get form field instances.

```
group = 'Fields'
```

```
integrate_with = 'drf_integration'
```

```
name = 'Checkbox select multiple'
```

```
submit_plugin_form_data(form_element_plugin, form_entry, request, serializer,
                        form_element_entries=None, **kwargs)
```

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** ([fobi.models.FormEntry](#)) – Instance of [fobi.models.FormEntry](#).
- **request** ([django.http.HttpRequest](#)) –
- **serializer** ([rest_framework.serializers.Serializer](#)) –
- **form_element_entries** ([iterable](#)) –

```
uid = 'checkbox_select_multiple'
```

48.4.1.1.1.70 Module contents

48.4.1.1.1.71 fobi.contrib.apps.drf_integration.form_elements.fields.date package

48.4.1.1.1.72 Submodules

48.4.1.1.1.73 fobi.contrib.apps.drf_integration.form_elements.fields.date.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date.apps.Config(app_name,
                                                                              app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_date'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.date'
```

48.4.1.1.1.74 fobi.contrib.apps.drf_integration.form_elements.fields.date.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date.base.DateSelectPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSubmitPluginFormDataMixin
    DateSelect select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Date'
    uid = 'date'
```

48.4.1.1.1.75 fobi.contrib.apps.drf_integration.form_elements.fields.date.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date.fobi_integration_form_elements.DateSe.
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSubmitPluginFormDataMixin
    DateSelect select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Date'
    uid = 'date'
```

48.4.1.1.1.76 Module contents

48.4.1.1.1.77 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down package

48.4.1.1.1.78 Submodules

48.4.1.1.1.79 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps.Config(app_name,
                                                                                       app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_date_drop_down'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down'
```

48.4.1.1.1.80 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.base.DateDropDownSelectPlug
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    DateSelect select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Date drop down'
    uid = 'date_drop_down'
```

48.4.1.1.1.81 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.fobi_integration_form_element module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.fobi_integration_form_eleme
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    DateSelect select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Date drop down'
    uid = 'date_drop_down'
```

48.4.1.1.1.82 Module contents

48.4.1.1.1.83 fobi.contrib.apps.drf_integration.form_elements.fields.datetime package

48.4.1.1.1.84 Submodules

48.4.1.1.1.85 fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps.Config(app_name,
                                                                                   app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_datetime'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.datetime'
```

48.4.1.1.1.86 fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base.DateTimeSelectPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    DateTimeSelect select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'DateTime'
    uid = 'datetime'
```

48.4.1.1.1.87 fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_integration_form_elements.DateTimeSelectPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    DateTimeSelect select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'DateTime'
    uid = 'datetime'
```

48.4.1.1.188 Module contents

48.4.1.1.189 fobi.contrib.apps.drf_integration.form_elements.fields.decimal package

48.4.1.1.190 Submodules

48.4.1.1.191 fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps.Config(app_name,
                                                                                   app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_decimal'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.decimal'
```

48.4.1.1.192 fobi.contrib.apps.drf_integration.form_elements.fields.decimal.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.decimal.base.DecimalInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    DecimalField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Decimal'
    uid = 'decimal'
```

48.4.1.1.193 fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi_integration_form_elements.Dec:
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    DecimalField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Decimal'
    uid = 'decimal'
```

48.4.1.1.1.94 Module contents

48.4.1.1.1.95 fobi.contrib.apps.drf_integration.form_elements.fields.duration package

48.4.1.1.1.96 Submodules

48.4.1.1.1.97 fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps.Config(app_name,
                                                                                   app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_duration'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.duration'
```

48.4.1.1.1.98 fobi.contrib.apps.drf_integration.form_elements.fields.duration.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.duration.base.DurationInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    DurationField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Duration'
    submit_plugin_form_data(form_element_plugin, form_entry, request, serializer,
                           form_element_entries=None, **kwargs)
        Submit plugin form data.
        Called on form submission (when user actually posts the data to assembled form).
    Parameters
        • form_element_plugin –
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • serializer (rest_framework.serializers.Serializer) –
        • form_element_entries (iterable) –
    uid = 'duration'
```

48.4.1.1.1.99 fobi.contrib.apps.drf_integration.form_elements.fields.duration.fobi_integration_form_elements module

class fobi.contrib.apps.drf_integration.form_elements.fields.duration.fobi_integration_form_elements.Du

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

DurationField plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*,
form_element_entries=None, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Duration'

submit_plugin_form_data(*form_element_plugin*, *form_entry*, *request*, *serializer*,
form_element_entries=None, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** ([fobi.models.FormEntry](#)) – Instance of [fobi.models.FormEntry](#).
- **request** ([django.http.HttpRequest](#)) –
- **serializer** ([rest_framework.serializers.Serializer](#)) –
- **form_element_entries** (*iterable*) –

uid = 'duration'

48.4.1.1.1.100 Module contents

48.4.1.1.1.101 fobi.contrib.apps.drf_integration.form_elements.fields.email package

48.4.1.1.1.102 Submodules

48.4.1.1.1.103 fobi.contrib.apps.drf_integration.form_elements.fields.email.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.email.apps.**Config**(*app_name*,
app_module)

Bases: [django.apps.config AppConfig](#)

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_email'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.email'

48.4.1.1.1.104 fobi.contrib.apps.drf_integration.form_elements.fields.email.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.email.base.EmailInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    EmailField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Decimal'
    uid = 'email'
```

48.4.1.1.1.105 fobi.contrib.apps.drf_integration.form_elements.fields.email.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.email.fobi_integration_form_elements.EmailInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    EmailField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Decimal'
    uid = 'email'
```

48.4.1.1.1.106 Module contents

48.4.1.1.1.107 fobi.contrib.apps.drf_integration.form_elements.fields.file package

48.4.1.1.1.108 Submodules

48.4.1.1.1.109 fobi.contrib.apps.drf_integration.form_elements.fields.file.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.file.apps.Config(app_name,
                                                                               app_module)
    Bases: django.apps.config AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_file'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.file'
```


48.4.1.1.110 fobi.contrib.apps.drf_integration.form_elements.fields.file.base module

class `fobi.contrib.apps.drf_integration.form_elements.fields.file.base.FileInputPlugin`(*user=None*)
 Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

FileField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'File'

submit_plugin_form_data(*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- **form_element_entries** (*iterable*) –

uid = 'file'

48.4.1.1.111 fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_integration_form_elements module

class `fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_integration_form_elements.FileInp`
 Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

FileField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'File'

submit_plugin_form_data(*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- `form_element_plugin` –
- `form_entry` (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- `request` (`django.http.HttpRequest`) –
- `serializer` (`rest_framework.serializers.Serializer`) –
- `form_element_entries` (`iterable`) –

`uid = 'file'`

48.4.1.1.1.112 Module contents

48.4.1.1.1.113 `fobi.contrib.apps.drf_integration.form_elements.fields.float` package

48.4.1.1.1.114 Submodules

48.4.1.1.1.115 `fobi.contrib.apps.drf_integration.form_elements.fields.float.apps` module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.float.apps.Config(app_name,
                                                                              app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_float'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.float'
```

48.4.1.1.1.116 `fobi.contrib.apps.drf_integration.form_elements.fields.float.base` module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.float.base.FloatInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    FloatField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Float'
    uid = 'float'
```

48.4.1.1.1.117 fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_integration_form_elements module

class fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_integration_form_elements.**Float**:

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

FloatField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Float'

uid = 'float'

48.4.1.1.1.118 Module contents

48.4.1.1.1.119 fobi.contrib.apps.drf_integration.form_elements.fields.hidden package

48.4.1.1.1.120 Submodules

48.4.1.1.1.121 fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps.**Config**(*app_name, app_module*)

Bases: [django.apps.config AppConfig](#)

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_hidden'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.hidden'

48.4.1.1.1.122 fobi.contrib.apps.drf_integration.form_elements.fields.hidden.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.hidden.base.**HiddenInputPlugin**(*user=None*)

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

CharField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Hidden'

uid = 'hidden'

48.4.1.1.1.123 fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration_form_elements.HiddenFormElementFieldPlugin(
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Hidden'
    uid = 'hidden'
```

48.4.1.1.1.124 Module contents

48.4.1.1.1.125 fobi.contrib.apps.drf_integration.form_elements.fields.input package

48.4.1.1.1.126 Submodules

48.4.1.1.1.127 fobi.contrib.apps.drf_integration.form_elements.fields.input.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.input.apps.Config(app_name,
                                                                               app_module)
    Bases: django.apps.config AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_input'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.input'
```

48.4.1.1.1.128 fobi.contrib.apps.drf_integration.form_elements.fields.input.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.input.base.InputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Text'
    uid = 'input'
```

48.4.1.1.1.129 fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration_form_elements module

class fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration_form_elements.**Input**

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

CharField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Text'

uid = 'input'

48.4.1.1.1.130 Module contents

48.4.1.1.1.131 fobi.contrib.apps.drf_integration.form_elements.fields.integer package

48.4.1.1.1.132 Submodules

48.4.1.1.1.133 fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps.**Config**(*app_name, app_module*)

Bases: [django.apps.config AppConfig](#)

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_integer'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.integer'

48.4.1.1.1.134 fobi.contrib.apps.drf_integration.form_elements.fields.integer.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.integer.base.**IntegerInputPlugin**(*user=None*)

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

IntegerField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Integer'

uid = 'integer'

48.4.1.1.1.135 fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_integration_form_elements.Int
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    IntegerField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Integer'
    uid = 'integer'
```

48.4.1.1.1.136 Module contents

48.4.1.1.1.137 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address package

48.4.1.1.1.138 Submodules

48.4.1.1.1.139 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps.Config(app_name,
                                           app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_ip_address'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.ip_address'
```

48.4.1.1.1.140 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base.IPAddressInputPlugin(user=N
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    IPAddressField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'IP address'
    uid = 'ip_address'
```

48.4.1.1.1.141 `fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration_form_elements` module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration_form_elements.  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.  
            DRFSubmitPluginFormDataMixin  
  
    IPAddressField plugin.  
  
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,  
                                form_element_entries=None, **kwargs)  
        Get form field instances.  
  
    group = 'Fields'  
    integrate_with = 'drf_integration'  
    name = 'IP address'  
    uid = 'ip_address'
```

48.4.1.1.1.142 Module contents

48.4.1.1.1.143 `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean` package

48.4.1.1.1.144 Submodules

48.4.1.1.1.145 `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps` module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps.Config(app_name,  
                                            app_module)  
  
    Bases: django.apps.config AppConfig  
    Config.  
  
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_null_boolean'  
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean'
```

48.4.1.1.1.146 `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.base` module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.base.NullBooleanSelectPlugin(  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.  
            DRFSubmitPluginFormDataMixin  
  
    NullBooleanField select plugin.  
  
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,  
                                form_element_entries=None, **kwargs)  
        Get form field instances.  
  
    group = 'Fields'  
    integrate_with = 'drf_integration'  
    name = 'Null boolean'  
    uid = 'null_boolean'
```

48.4.1.1.1.147 fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form_elements module

class fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form_elements.

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

NullBooleanField select plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*,
form_element_entries=None, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Null boolean'

uid = 'null_boolean'

48.4.1.1.1.148 Module contents

48.4.1.1.1.149 fobi.contrib.apps.drf_integration.form_elements.fields.password package

48.4.1.1.1.150 Submodules

48.4.1.1.1.151 fobi.contrib.apps.drf_integration.form_elements.fields.password.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.password.apps.**Config**(*app_name*,
app_module)

Bases: [django.apps.config AppConfig](#)

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_password'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.password'

48.4.1.1.1.152 fobi.contrib.apps.drf_integration.form_elements.fields.password.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.password.base.**PasswordInputPlugin**(*user=None*)

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

CharField plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*,
form_element_entries=None, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Text'

uid = 'password'

48.4.1.1.1.153 `fobi.contrib.apps.drf_integration.form_elements.fields.password.fobi_integration_form_elements` module

class `fobi.contrib.apps.drf_integration.form_elements.fields.password.fobi_integration_form_elements.PasswordFormElementPlugin`

Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

CharField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Text'

uid = 'password'

48.4.1.1.1.154 Module contents

48.4.1.1.1.155 `fobi.contrib.apps.drf_integration.form_elements.fields.radio` package

48.4.1.1.1.156 Submodules

48.4.1.1.1.157 `fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps` module

class `fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps.Config`(*app_name, app_module*)

Bases: `django.apps.config AppConfig`

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_radio'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.radio'

48.4.1.1.1.158 `fobi.contrib.apps.drf_integration.form_elements.fields.radio.base` module

class `fobi.contrib.apps.drf_integration.form_elements.fields.radio.base.RadioInputPlugin`(*user=None*)

Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

ChoiceField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Radio'

uid = 'radio'

48.4.1.1.1.159 fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_integration_form_elements module

class fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_integration_form_elements.**Radio**

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

ChoiceField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Radio'

uid = 'radio'

48.4.1.1.1.160 Module contents

48.4.1.1.1.161 fobi.contrib.apps.drf_integration.form_elements.fields.range_select package

48.4.1.1.1.162 Submodules

48.4.1.1.1.163 fobi.contrib.apps.drf_integration.form_elements.fields.range_select.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.range_select.apps.**Config**(*app_name, app_module*)

Bases: [django.apps.config AppConfig](#)

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_range_select'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.range_select'

48.4.1.1.1.164 fobi.contrib.apps.drf_integration.form_elements.fields.range_select.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.range_select.base.**RangeSelectInputPlugin**(*u*

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

ChoiceField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Range select'

uid = 'range_select'

48.4.1.1.1.165 fobi.contrib.apps.drf_integration.form_elements.fields.range_select.fobi_integration_form_elements module

class fobi.contrib.apps.drf_integration.form_elements.fields.range_select.fobi_integration_form_elements.

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

ChoiceField plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Range select'

uid = 'range_select'

48.4.1.1.1.166 Module contents

48.4.1.1.1.167 fobi.contrib.apps.drf_integration.form_elements.fields.regex package

48.4.1.1.1.168 Submodules

48.4.1.1.1.169 fobi.contrib.apps.drf_integration.form_elements.fields.regex.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.regex.apps.**Config**(*app_name*, *app_module*)

Bases: [django.apps.config AppConfig](#)

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_regex'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.regex'

48.4.1.1.1.170 fobi.contrib.apps.drf_integration.form_elements.fields.regex.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.regex.base.**RegexInputPlugin**(*user=None*)

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

RegexField plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Regex'

uid = 'regex'

48.4.1.1.1.171 fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_integration_form_elements.RegexFieldPlugin(
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    RegexField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Regex'
    uid = 'regex'
```

48.4.1.1.1.172 Module contents

48.4.1.1.1.173 fobi.contrib.apps.drf_integration.form_elements.fields.select package

48.4.1.1.1.174 Submodules

48.4.1.1.1.175 fobi.contrib.apps.drf_integration.form_elements.fields.select.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select.apps.Config(app_name,
                                                                                app_module)
    Bases: django.apps.config.AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select'
```

48.4.1.1.1.176 fobi.contrib.apps.drf_integration.form_elements.fields.select.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select.base.SelectInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Select'
    uid = 'select'
```

48.4.1.1.1.177 fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi_integration_form_elements.SelectModelObject:
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin

    ChoiceField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)

        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Select'
    uid = 'select'
```

48.4.1.1.1.178 Module contents

48.4.1.1.1.179 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object package

48.4.1.1.1.180 Submodules

48.4.1.1.1.181 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.apps.Config(app_name,
                                                                                             app_module):
    Bases: django.apps.config AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select_model_object'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object'
```

48.4.1.1.1.182 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.base.SelectModelObject:
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin

    ChoiceField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)

        Get form field instances.

    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Select model object'
```

submit_plugin_form_data(*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_model_object'

48.4.1.1.183 *fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.fobi_integration_form_e* module

class *fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.fobi_integration_form_e*
 Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

ChoiceField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Select model object'

submit_plugin_form_data(*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_model_object'

48.4.1.1.1.184 Module contents

48.4.1.1.1.185 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple package

48.4.1.1.1.186 Submodules

48.4.1.1.1.187 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.apps.Config(app_name,
                                                                                       app_module)
```

Bases: `django.apps.config AppConfig`

`Config`.

`label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select_multiple'`

`name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple'`

48.4.1.1.1.188 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base.SelectMultipleInputPl
```

Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

MultipleChoiceField plugin.

```
get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                           form_element_entries=None, **kwargs)
```

Get form field instances.

`group = 'Fields'`

`integrate_with = 'drf_integration'`

`name = 'Select multiple'`

```
submit_plugin_form_data(form_element_plugin, form_entry, request, serializer,
                        form_element_entries=None, **kwargs)
```

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- `form_element_plugin` –
- `form_entry` (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- `request` (`django.http.HttpRequest`) –
- `serializer` (`rest_framework.serializers.Serializer`) –
- `form_element_entries` (`iterable`) –

`uid = 'select_multiple'`

48.4.1.1.1.189 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.fobi_integration_form_element module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.fobi_integration_form_element

Bases: [fobi.base.IntegrationFormFieldPlugin](#), [fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin](#)

MultipleChoiceField plugin.

get_custom_field_instances(*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Select multiple'

submit_plugin_form_data(*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** ([fobi.models.FormEntry](#)) – Instance of [fobi.models.FormEntry](#).
- **request** ([django.http.HttpRequest](#)) –
- **serializer** ([rest_framework.serializers.Serializer](#)) –
- **form_element_entries** (*iterable*) –

uid = 'select_multiple'

48.4.1.1.1.190 Module contents

48.4.1.1.1.191 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects package

48.4.1.1.1.192 Submodules

48.4.1.1.1.193 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.apps.Config

Bases: [django.apps.config AppConfig](#)

Config.

label =

'fobi_contrib_apps_drf_integration_form_elements_fields_select_multiple_model_objects'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects'

48.4.1.1.194 `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.base` module

class `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.base.SelectMultipleModelChoiceFieldPlugin`

Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

ModelMultipleChoiceField plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Select multiple model objects'

submit_plugin_form_data(*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- **form_element_entries** (*iterable*) –

uid = 'select_multiple_model_objects'

48.4.1.1.195 `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.fobi_integration` module

class `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.fobi_integration`

Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

ModelMultipleChoiceField plugin.

get_custom_field_instances(*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Select multiple model objects'

submit_plugin_form_data(*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- `form_element_plugin` –
- `form_entry` (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- `request` (`django.http.HttpRequest`) –
- `serializer` (`rest_framework.serializers.Serializer`) –
- `form_element_entries` (`iterable`) –

`uid = 'select_multiple_model_objects'`

48.4.1.1.1.196 Module contents

48.4.1.1.1.197 `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max` package

48.4.1.1.1.198 Submodules

48.4.1.1.1.199 `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.apps` module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.apps.Config(app_name='fobi',
app_name='fobi')
```

Bases: `django.apps.config AppConfig`

Config.

```
label =
'fobi_contrib_apps_drf_integration_form_elements_fields_select_multiple_with_max'
name =
'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max'
```

48.4.1.1.1.200 `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.base` module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.base.SelectMultipleWithMaxFieldPlugin:
```

Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

MultipleChoiceField with max plugin.

```
get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
form_element_entries=None, **kwargs)
```

Get form field instances.

```
group = 'Fields'
```

```
integrate_with = 'drf_integration'
```

```
name = 'Select multiple with max'
```

```
submit_plugin_form_data(form_element_plugin, form_entry, request, serializer,
form_element_entries=None, **kwargs)
```

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- **form_element_entries** (`iterable`) –

`uid = 'select_multiple_with_max'`

48.4.1.1.1.201 `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.fobi_integration_fobi_module`

`class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.fobi_integration_fobi_module`

Bases: `fobi.base.IntegrationFormFieldPlugin`, `fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin`

MultipleChoiceField with max plugin.

get_custom_field_instances(`form_element_plugin`, `request=None`, `form_entry=None`, `form_element_entries=None`, `**kwargs`)

Get form field instances.

group = 'Fields'

integrate_with = 'drf_integration'

name = 'Select multiple with max'

submit_plugin_form_data(`form_element_plugin`, `form_entry`, `request`, `serializer`, `form_element_entries=None`, `**kwargs`)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- **form_element_entries** (`iterable`) –

`uid = 'select_multiple_with_max'`

48.4.1.1.1.202 Module contents

48.4.1.1.1.203 fobi.contrib.apps.drf_integration.form_elements.fields.slider package

48.4.1.1.1.204 Submodules

48.4.1.1.1.205 fobi.contrib.apps.drf_integration.form_elements.fields.slider.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slider.apps.Config(app_name,
                                                                               app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_slider'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.slider'
```

48.4.1.1.1.206 fobi.contrib.apps.drf_integration.form_elements.fields.slider.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slider.base.SliderInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Slider'
    uid = 'slider'
```

48.4.1.1.1.207 fobi.contrib.apps.drf_integration.form_elements.fields.slider.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slider.fobi_integration_form_elements.SliderInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Slider'
    uid = 'slider'
```

48.4.1.1.1.208 Module contents

48.4.1.1.1.209 fobi.contrib.apps.drf_integration.form_elements.fields.slug package

48.4.1.1.1.210 Submodules

48.4.1.1.1.211 fobi.contrib.apps.drf_integration.form_elements.fields.slug.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slug.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_slug'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.slug'
```

48.4.1.1.1.212 fobi.contrib.apps.drf_integration.form_elements.fields.slug.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slug.base.SlugInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    SlugField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Slug'
    uid = 'slug'
```

48.4.1.1.1.213 fobi.contrib.apps.drf_integration.form_elements.fields.slug.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slug.fobi_integration_form_elements.SlugInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    SlugField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Slug'
    uid = 'slug'
```

48.4.1.1.1.214 Module contents

48.4.1.1.1.215 fobi.contrib.apps.drf_integration.form_elements.fields.text package

48.4.1.1.1.216 Submodules

48.4.1.1.1.217 fobi.contrib.apps.drf_integration.form_elements.fields.text.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.text.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_text'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.text'
```

48.4.1.1.1.218 fobi.contrib.apps.drf_integration.form_elements.fields.text.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.text.base.TextInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Text'
    uid = 'text'
```

48.4.1.1.1.219 fobi.contrib.apps.drf_integration.form_elements.fields.text.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.text.fobi_integration_form_elements.TextIn
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Text'
    uid = 'text'
```

48.4.1.1.1.220 Module contents

48.4.1.1.1.221 fobi.contrib.apps.drf_integration.form_elements.fields.textarea package

48.4.1.1.1.222 Submodules

48.4.1.1.1.223 fobi.contrib.apps.drf_integration.form_elements.fields.textarea.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.textarea.apps.Config(app_name,
                                                                                   app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_textarea'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.textarea'
```

48.4.1.1.1.224 fobi.contrib.apps.drf_integration.form_elements.fields.textarea.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.textarea.base.TextareaPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Textarea'
    uid = 'textarea'
```

48.4.1.1.1.225 fobi.contrib.apps.drf_integration.form_elements.fields.textarea.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.textarea.fobi_integration_form_elements.Te
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
           DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Textarea'
    uid = 'textarea'
```

48.4.1.1.1.226 Module contents

48.4.1.1.1.227 fobi.contrib.apps.drf_integration.form_elements.fields.time package

48.4.1.1.1.228 Submodules

48.4.1.1.1.229 fobi.contrib.apps.drf_integration.form_elements.fields.time.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.time.apps.Config(app_name,
                                                                              app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_time'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.time'
```

48.4.1.1.1.230 fobi.contrib.apps.drf_integration.form_elements.fields.time.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.time.base.TimeInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    TimeField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Time'
    uid = 'time'
```

48.4.1.1.1.231 fobi.contrib.apps.drf_integration.form_elements.fields.time.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.time.fobi_integration_form_elements.TimeInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    TimeField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'Time'
    uid = 'time'
```


48.4.1.1.1.232 Module contents

48.4.1.1.1.233 fobi.contrib.apps.drf_integration.form_elements.fields.url package

48.4.1.1.1.234 Submodules

48.4.1.1.1.235 fobi.contrib.apps.drf_integration.form_elements.fields.url.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.url.apps.Config(app_name,
                                                                           app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_url'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.url'
```

48.4.1.1.1.236 fobi.contrib.apps.drf_integration.form_elements.fields.url.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.url.base.URLInputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    URLField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'URL'
    uid = 'url'
```

48.4.1.1.1.237 fobi.contrib.apps.drf_integration.form_elements.fields.url.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.url.fobi_integration_form_elements.URLInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.
    DRFSSubmitPluginFormDataMixin
    URLField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    integrate_with = 'drf_integration'
    name = 'URL'
    uid = 'url'
```

48.4.1.1.1.238 Module contents

48.4.1.1.1.239 Module contents

48.4.1.1.1.240 Module contents

48.4.1.1.1.241 fobi.contrib.apps.drf_integration.form_handlers package

48.4.1.1.1.242 Subpackages

48.4.1.1.1.243 fobi.contrib.apps.drf_integration.form_handlers.db_store package

48.4.1.1.1.244 Submodules

48.4.1.1.1.245 fobi.contrib.apps.drf_integration.form_handlers.db_store.apps module

```
class fobi.contrib.apps.drf_integration.form_handlers.db_store.apps.Config(app_name,
                                                                           app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_handlers_db_store'
    name = 'fobi.contrib.apps.drf_integration.form_handlers.db_store'
```

48.4.1.1.1.246 fobi.contrib.apps.drf_integration.form_handlers.db_store.base module

```
class fobi.contrib.apps.drf_integration.form_handlers.db_store.base.DBStoreHandlerPlugin(user=None)
    Bases: fobi.base.IntegrationFormHandlerPlugin
    DB store form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
    name = 'DB store'
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'db_store'
```

48.4.1.1.1.247 fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration_form_handlers module

```
class fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration_form_handlers.DBStoreHandlerPlugin(user=None)
    Bases: fobi.base.IntegrationFormHandlerPlugin
    DB store form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
```

```
name = 'DB store'

run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
    Run.

uid = 'db_store'
```

48.4.1.1.1.248 Module contents

48.4.1.1.1.249 fobi.contrib.apps.drf_integration.form_handlers.http_repost package

48.4.1.1.1.250 Submodules

48.4.1.1.1.251 fobi.contrib.apps.drf_integration.form_handlers.http_repost.apps module

```
class fobi.contrib.apps.drf_integration.form_handlers.http_repost.apps.Config(app_name,
                                                                              app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_handlers_http_repost'
    name = 'fobi.contrib.apps.drf_integration.form_handlers.http_repost'
```

48.4.1.1.1.252 fobi.contrib.apps.drf_integration.form_handlers.http_repost.base module

```
class fobi.contrib.apps.drf_integration.form_handlers.http_repost.base.HTTPRepostHandlerPlugin(user=None)
    Bases: fobi.base.IntegrationFormHandlerPlugin
    HTTP re-post handler plugin.
    integrate_with = 'drf_integration'
    name = 'Mail'
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'http_repost'
```

48.4.1.1.1.253 fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handlers module

```
class fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handlers.HTTPRepostHandlerPlugin(user=None)
    Bases: fobi.base.IntegrationFormHandlerPlugin
    HTTP re-post handler plugin.
    integrate_with = 'drf_integration'
    name = 'Mail'
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'http_repost'
```

48.4.1.1.1.254 Module contents

48.4.1.1.1.255 fobi.contrib.apps.drf_integration.form_handlers.mail package

48.4.1.1.1.256 Submodules

48.4.1.1.1.257 fobi.contrib.apps.drf_integration.form_handlers.mail.apps module

```
class fobi.contrib.apps.drf_integration.form_handlers.mail.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_handlers_mail'
    name = 'fobi.contrib.apps.drf_integration.form_handlers.mail'
```

48.4.1.1.1.258 fobi.contrib.apps.drf_integration.form_handlers.mail.base module

```
class fobi.contrib.apps.drf_integration.form_handlers.mail.base.MailHandlerPlugin(user=None)
    Bases: fobi.base.IntegrationFormHandlerPlugin
    Mail handler form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
    name = 'Mail'
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'mail'
```

48.4.1.1.1.259 fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_integration_form_handlers module

```
class fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_integration_form_handlers.MailHandlerPl
    Bases: fobi.base.IntegrationFormHandlerPlugin
    Mail handler form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
    name = 'Mail'
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'mail'
```

48.4.1.1.1.260 Module contents

48.4.1.1.1.261 Module contents

48.4.1.1.1.262 Submodules

48.4.1.1.1.263 fobi.contrib.apps.drf_integration.apps module

```
class fobi.contrib.apps.drf_integration.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration'
    name = 'fobi.contrib.apps.drf_integration'
```

48.4.1.1.1.264 fobi.contrib.apps.drf_integration.base module

```
class fobi.contrib.apps.drf_integration.base.DRFIntegrationFormElementPluginProcessor(*args,
                                                                                      **kwargs)
```

Bases: *fobi.base.IntegrationFormElementPluginProcessor*

Django REST framework field instance processor.

```
process_custom_form_field_instance(form_element_entry, form_entry, request,
                                   form_element_plugin=None)
```

Process.

```
class fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
```

Bases: object

Submit plugin form data mixin.

```
submit_plugin_form_data(form_element_plugin, form_entry, request, serializer,
                        form_element_entries=None, **kwargs)
```

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

```
fobi.contrib.apps.drf_integration.base.get_cleaned_data(serializer, keys_to_remove=[],
                                                         values_to_remove=[])
```

Get cleaned data.

Gets cleaned data, having the trash (fields without values) filtered out.

Parameters

- **serializer** –

- `keys_to_remove(iterable)` –
- `values_to_remove(iterable)` –

Return dict

```
fobi.contrib.apps.drf_integration.base.get_field_name_to_label_map(serializer,
                                                                    keys_to_remove=[],
                                                                    values_to_remove=[])
```

Get field name to label map.

Parameters

- **serializer** –
- `keys_to_remove(iterable)` –
- `values_to_remove(iterable)` –

Return dict

```
fobi.contrib.apps.drf_integration.base.get_processed_serializer_data(serializer,
                                                                    form_element_entries)
```

Gets processed serializer data.

Simply fires both `fobi.base.get_cleaned_data` and `fobi.base.get_field_name_to_label_map` functions and returns the result.

Parameters

- **serializer** –
- `form_element_entries(iterable)` – Iterable of form element entries.

Return tuple

```
fobi.contrib.apps.drf_integration.base.run_form_handlers(form_entry, request, serializer,
                                                         form_element_entries=None)
```

Run form handlers.

Parameters

- **form_entry** (`fobi.models.FormEntry`) –
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- `form_element_entries(iterable)` –

Return tuple List of success responses, list of error responses

```
fobi.contrib.apps.drf_integration.base.submit_plugin_form_data(form_entry, request, serializer,
                                                             form_element_entries=None,
                                                             **kwargs)
```

Submit plugin form data for all plugins.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- `form_element_entries(iterable)` –

48.4.1.1.1.265 fobi.contrib.apps.drf_integration.dynamic module

`fobi.contrib.apps.drf_integration.dynamic.assemble_serializer_class`(*form_entry*, *request=None*, *origin=None*, *origin_kwargs_update_func=None*, *origin_return_func=None*, *form_element_entries=None*, *has_value=None*, *declared_fields=None*, *declared_fields_metadata=None*)

Assemble a serializer class by given entry.

Parameters

- **form_entry** –
- **base_class** –
- **request** (*django.http.HttpRequest*) –
- **origin** (*string*) –
- **origin_kwargs_update_func** (*callable*) –
- **origin_return_func** (*callable*) –
- **form_element_entries** (*iterable*) – If given, used instead of `form_entry`. `formelemententry_set.all` (no additional database hit).
- **has_value** (*bool*) –

`fobi.contrib.apps.drf_integration.dynamic.get_declared_fields`(*form_entry*, *request=None*, *origin=None*, *origin_kwargs_update_func=None*, *origin_return_func=None*, *form_element_entries=None*, *has_value=None*)

Get declared fields.

48.4.1.1.1.266 fobi.contrib.apps.drf_integration.fields module

class `fobi.contrib.apps.drf_integration.fields.ContentImageField`(*args, **kwargs)

Bases: `fobi.contrib.apps.drf_integration.fields.NoneField`

Content image field.

class `fobi.contrib.apps.drf_integration.fields.ContentMarkdownField`(*args, **kwargs)

Bases: `fobi.contrib.apps.drf_integration.fields.NoneField`

Content markdown field.

class `fobi.contrib.apps.drf_integration.fields.ContentRichTextField`(*args, **kwargs)

Bases: `fobi.contrib.apps.drf_integration.fields.NoneField`

Content rich text field.

class `fobi.contrib.apps.drf_integration.fields.ContentTextField`(*args, **kwargs)

Bases: `fobi.contrib.apps.drf_integration.fields.NoneField`

Content text field.

```
class fobi.contrib.apps.drf_integration.fields.ContentVideoField(*args, **kwargs)
    Bases: fobi.contrib.apps.drf_integration.fields.NoneField

    Content video field.

class fobi.contrib.apps.drf_integration.fields.ModelChoiceField(*args, **kwargs)
    Bases: rest_framework.fields.ChoiceField, fobi.contrib.apps.drf_integration.fields.
    ModelChoiceFieldMixin

    Model choice field.

class fobi.contrib.apps.drf_integration.fields.ModelMultipleChoiceField(*args, **kwargs)
    Bases: rest_framework.fields.MultipleChoiceField, fobi.contrib.apps.drf_integration.
    fields.ModelChoiceFieldMixin

    Model choice field.

class fobi.contrib.apps.drf_integration.fields.MultipleChoiceWithMaxField(*args, **kwargs)
    Bases: rest_framework.fields.MultipleChoiceField

    MultipleChoiceWithMaxField.

    default_error_messages = {'empty': 'This selection may not be empty.',
    'invalid_choice': '"{input}" is not a valid choice.', 'max_choices': 'Max number
    of choices reached.', 'not_a_list': 'Expected a list of items but got type
    "{input_type}".'}

    to_internal_value(data)
        Transform the incoming primitive data into a native value.

class fobi.contrib.apps.drf_integration.fields.NoneField(*args, **kwargs)
    Bases: rest_framework.fields.Field

    NoneField.

    default_empty_html = ''

    default_error_messages = {}

    initial = ''

    run_validation(data=<class 'rest_framework.fields.empty'>)
        Validate a simple representation and return the internal value.

        The provided data may be empty if no representation was included in the input.

        May raise SkipField if the field should not be included in the validated data.

    to_internal_value(data)
        Transform the incoming primitive data into a native value.

    to_representation(value)
        Transform the outgoing native value into primitive data.
```


48.4.1.1.1.267 fobi.contrib.apps.drf_integration.helpers module

`fobi.contrib.apps.drf_integration.helpers.map_field_name_to_label(serializer)`

Takes a form and creates label to field name map.

Parameters `serializer` – Instance of `rest_framework.serializers.Serializer`.

Return dict

48.4.1.1.1.268 fobi.contrib.apps.drf_integration.metadata module

class `fobi.contrib.apps.drf_integration.metadata.FobiMetaData`

Bases: `rest_framework.metadata.SimpleMetadata`

Meta data for better representation of the form elements.

get_field_info(*field*)

Get field info.

Given an instance of a serializer field, return a dictionary of metadata about it.

label_lookup = `<rest_framework.utils.field_mapping.ClassLookupDict object>`

48.4.1.1.1.269 fobi.contrib.apps.drf_integration.serializers module

class `fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer(*args, **kwargs)`

Bases: `rest_framework.serializers.ModelSerializer`

FormEntry serializer (read-only).

Used for detail and list views.

class Meta

Bases: `object`

Options.

fields = ('url', 'id', 'slug', 'title')

model

alias of `fobi.models.FormEntry`

read_only_fields = ('url', 'id', 'slug', 'title')

get_fields_metadata(*field_name=None*)

Just to make sure nothing breaks.

get_url(*obj*)

Get URL.

48.4.1.1.1.270 fobi.contrib.apps.drf_integration.urls module

48.4.1.1.1.271 fobi.contrib.apps.drf_integration.utils module

`fobi.contrib.apps.drf_integration.utils.get_serializer_class`(*form_entry*, *request=None*,
has_value=None,
declared_fields=None)

Get assembled serializer class.

Parameters

- **form_entry** (`fobi.models.FormEntry`) –
- **request** (`django.http.HttpRequest`) –
- **has_value** (`bool`) –
- **declared_fields** (`list`) –

Return `django.forms.Form`

48.4.1.1.1.272 fobi.contrib.apps.drf_integration.views module

```
class fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet(**kwargs)
    Bases: rest_framework.mixins.RetrieveModelMixin, rest_framework.mixins.
    UpdateModelMixin, rest_framework.mixins.ListModelMixin, rest_framework.viewsets.
    GenericViewSet
    FormEntry view set.
    basename = None
    description = None
    detail = None
    get_object()
        Override get_object to get things done.
    get_queryset()
        Get queryset.
        We show all forms to authenticated users and show only public forms to non-authenticated users.
    get_serializer(*args, **kwargs)
        Get the serializer.
    get_serializer_class()
        Get serializer class.
    has_value()
    lookup_field = 'slug'
    lookup_url_kwarg = 'slug'
    metadata_class
        alias of fobi.contrib.apps.drf_integration.metadata.FobiMetaData
    name = None
    permission_classes = [<class 'rest_framework.permissions.AllowAny'>]
```

```

queryset
suffix = None
update(request, *args, **kwargs)

```

48.4.1.1.1.273 Module contents

48.4.1.1.1.274 fobi.contrib.apps.feincms_integration package

48.4.1.1.1.275 Submodules

48.4.1.1.1.276 fobi.contrib.apps.feincms_integration.apps module

```

class fobi.contrib.apps.feincms_integration.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_feincms_integration'
    name = 'fobi.contrib.apps.feincms_integration'

```

48.4.1.1.1.277 fobi.contrib.apps.feincms_integration.conf module

```

fobi.contrib.apps.feincms_integration.conf.get_setting(setting, override=None)
    Get setting.

    Get a setting from fobi.contrib.apps.feincms_integration conf module, falling back to the default.

    If override is not None, it will be used instead of the setting.

```

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.278 fobi.contrib.apps.feincms_integration.defaults module

48.4.1.1.1.279 fobi.contrib.apps.feincms_integration.helpers module

```

fobi.contrib.apps.feincms_integration.helpers.get_form_template_choices()
    Gets the form template choices.

```

It's possible to provide theme templates per theme or just per project.

Return list

```

fobi.contrib.apps.feincms_integration.helpers.get_success_page_template_choices()
    Get success page template choices.

```

Return list

48.4.1.1.1.280 fobi.contrib.apps.feincms_integration.settings module

- `WIDGET_FORM_SENT_GET_PARAM` (str): Name of the GET param indicating that form has been successfully sent.

48.4.1.1.1.281 fobi.contrib.apps.feincms_integration.widgets module

class `fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget(*args, **kwargs)`
 Bases: `django.db.models.base.Model`, `fobi.integration.processors.IntegrationProcessor`

Widget for to FeinCMS.

Property `fobi.models.FormEntry form_entry` Form entry to be rendered.

Property `str template` If given used for rendering the form.

class `Meta`

Bases: `object`

Meta options.

abstract = `False`

app_label = `'fobi'`

can_redirect = `True`

finalize(*request, response*)

Finalize.

form_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

form_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

form_sent_get_param = `'sent'`

form_submit_button_text

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

form_template_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

form_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_form_template_name_display(**, field=<django.db.models.fields.CharField: form_template_name>*)

get_success_page_template_name_display(**, field=<django.db.models.fields.CharField: success_page_template_name>*)

hide_form_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

hide_success_page_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

process(*request, **kwargs*)

This is where most of the form handling happens.

Parameters *request* (*django.http.HttpRequest*) –

Return *django.http.HttpResponse* | *str*

render(**kwargs*)

Render.

success_page_template_name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

success_page_text

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

success_page_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

48.4.1.1.1.282 Module contents

48.4.1.1.1.283 fobi.contrib.apps.mezzanine_integration package

48.4.1.1.1.284 Submodules

48.4.1.1.1.285 fobi.contrib.apps.mezzanine_integration.admin module

48.4.1.1.1.286 fobi.contrib.apps.mezzanine_integration.apps module

class *fobi.contrib.apps.mezzanine_integration.apps.Config*(*app_name, app_module*)

Bases: *django.apps.config AppConfig*

Config.

label = *'fobi_contrib_apps_mezzanine_integration'*

name = *'fobi.contrib.apps.mezzanine_integration'*

48.4.1.1.1.287 fobi.contrib.apps.mezzanine_integration.conf module

`fobi.contrib.apps.mezzanine_integration.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from `fobi.contrib.apps.mezzanine_integration` conf module, falling back to the default.

If `override` is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

48.4.1.1.1.288 fobi.contrib.apps.mezzanine_integration.defaults module

48.4.1.1.1.289 fobi.contrib.apps.mezzanine_integration.helpers module

`fobi.contrib.apps.mezzanine_integration.helpers.get_form_template_choices()`

Gets the form template choices.

It's possible to provide theme templates per theme or just per project.

Return list

`fobi.contrib.apps.mezzanine_integration.helpers.get_success_page_template_choices()`

Get success page template choices.

Return list

48.4.1.1.1.290 fobi.contrib.apps.mezzanine_integration.models module

48.4.1.1.1.291 fobi.contrib.apps.mezzanine_integration.page_processors module

48.4.1.1.1.292 fobi.contrib.apps.mezzanine_integration.settings module

- `WIDGET_FORM_SENT_GET_PARAM` (str): Name of the GET param indicating that form has been successfully sent.

48.4.1.1.1.293 Module contents

48.4.1.1.1.294 Module contents

48.4.1.1.1.295 fobi.contrib.plugins package

48.4.1.1.1.296 Subpackages

48.4.1.1.1.297 fobi.contrib.plugins.form_elements package

48.4.1.1.1.298 Subpackages

48.4.1.1.1.299 fobi.contrib.plugins.form_elements.content package

48.4.1.1.1.300 Subpackages

48.4.1.1.1.301 fobi.contrib.plugins.form_elements.content.content_image package

48.4.1.1.1.302 Submodules

48.4.1.1.1.303 fobi.contrib.plugins.form_elements.content.content_image.apps module

```
class fobi.contrib.plugins.form_elements.content.content_image.apps.Config(app_name,
                                                                           app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_content_content_image'
    name = 'fobi.contrib.plugins.form_elements.content.content_image'
```

48.4.1.1.1.304 fobi.contrib.plugins.form_elements.content.content_image.base module

```
class fobi.contrib.plugins.form_elements.content.content_image.base.ContentImagePlugin(user=None)
    Bases: fobi.base.FormElementPlugin
    Content image plugin.
    clone_plugin_data(entry)
        Clone plugin data.
        Clone plugin data, which means we make a copy of the original image.
        TODO: Perhaps rely more on data of form_element_entry?
    delete_plugin_data()
        Delete uploaded file.
    form
        alias of fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm
```

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
Get form field instances.

get_raw_data()
Get raw data.

Might be used in integration plugins.

get_rendered_image()
Get rendered image.

group = 'Content'

name = 'Content image'

post_processor()
Post process data.

Always the same.

uid = 'content_image'

48.4.1.1.1.305 fobi.contrib.plugins.form_elements.content.content_image.conf module

fobi.contrib.plugins.form_elements.content.content_image.conf.get_setting(*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.content.content_image.conf` module, falling back to the default.

If `override` is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

48.4.1.1.1.306 fobi.contrib.plugins.form_elements.content.content_image.defaults module

48.4.1.1.1.307 fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements.ContentImagePlugin`(*use*)
Bases: `fobi.base.FormElementPlugin`

Content image plugin.

clone_plugin_data(*entry*)
Clone plugin data.

Clone plugin data, which means we make a copy of the original image.

TODO: Perhaps rely more on data of `form_element_entry`?

delete_plugin_data()
Delete uploaded file.


```

form
    alias      of      fobi.contrib.plugins.form_elements.content.content_image.forms.
                        ContentImageForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
    Get form field instances.

get_raw_data()
    Get raw data.

    Might be used in integration plugins.

get_rendered_image()
    Get rendered image.

group = 'Content'
name = 'Content image'

post_processor()
    Post process data.

    Always the same.

uid = 'content_image'

```

48.4.1.1.1.308 fobi.contrib.plugins.form_elements.content.content_image.forms module

```

class fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm(data=None,
                                                                 files=None,
                                                                 auto_id='id_%s',
                                                                 pre-
                                                                 fix=None,
                                                                 ini-
                                                                 tial=None,
                                                                 er-
                                                                 ror_class=<class
                                                                 'django.forms.utils.Err
                                                                 la-
                                                                 bel_suffix=None,
                                                                 empty_permitted=False,
                                                                 field_order=None,
                                                                 use_required_attribute
                                                                 ren-
                                                                 derer=None)

```

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

Form for ContentImagePlugin.

```

base_fields = {'alt': <django.forms.fields.CharField object>, 'file':
<django.forms.fields.ImageField object>, 'fit_method':
<django.forms.fields.ChoiceField object>, 'size': <django.forms.fields.ChoiceField
object>}

declared_fields = {'alt': <django.forms.fields.CharField object>, 'file':
<django.forms.fields.ImageField object>, 'fit_method':
<django.forms.fields.ChoiceField object>, 'size': <django.forms.fields.ChoiceField
object>}

```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('file', ''), ('alt', ''), ('fit_method', 'center'), ('size',
'500x500')]
```

```
save_plugin_data(request=None)
```

Saving the plugin data and moving the file.

48.4.1.1.1.309 fobi.contrib.plugins.form_elements.content.content_image.helpers module

```
fobi.contrib.plugins.form_elements.content.content_image.helpers.get_crop_filter(fit_method)
```

Get crop filter.

48.4.1.1.1.310 fobi.contrib.plugins.form_elements.content.content_image.settings module

- FIT_METHOD_CROP_SMART (string)
- FIT_METHOD_CROP_CENTER (string)
- FIT_METHOD_CROP_SCALE (string)
- FIT_METHOD_FIT_WIDTH (string)
- FIT_METHOD_FIT_HEIGHT (string)
- DEFAULT_FIT_METHOD (string)
- FIT_METHODS_CHOICES (tuple)
- FIT_METHODS_CHOICES_WITH_EMPTY_OPTION (list)
- IMAGES_UPLOAD_DIR (string)

48.4.1.1.1.311 Module contents

48.4.1.1.1.312 fobi.contrib.plugins.form_elements.content.content_image_url package

48.4.1.1.1.313 Submodules

48.4.1.1.1.314 fobi.contrib.plugins.form_elements.content.content_image_url.apps module

```
class fobi.contrib.plugins.form_elements.content.content_image_url.apps.Config(app_name,
                                                                              app_module)
```

Bases: django.apps.config AppConfig

Config.

```
label = 'fobi_contrib_plugins_form_elements_content_content_image_url'
```

```
name = 'fobi.contrib.plugins.form_elements.content.content_image_url'
```

48.4.1.1.1.315 fobi.contrib.plugins.form_elements.content.content_image_url.base module

class `fobi.contrib.plugins.form_elements.content.content_image_url.base.ContentImageURLPlugin`(*user=None*)

Bases: `fobi.base.FormElementPlugin`

Content image plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_image_url.forms.ContentImageURLForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

get_raw_data()

Get raw data.

Might be used in integration plugins.

get_rendered_image()

Get rendered image.

group = 'Content'

name = 'Content image URL'

post_processor()

Post process data.

Always the same.

uid = 'content_image_url'

48.4.1.1.1.316 fobi.contrib.plugins.form_elements.content.content_image_url.conf module

`fobi.contrib.plugins.form_elements.content.content_image_url.conf.get_setting`(*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.content.content_image_url` conf module, falling back to the default.

If *override* is not *None*, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to *None*.

Returns Setting value.

48.4.1.1.1.317 fobi.contrib.plugins.form_elements.content.content_image_url.defaults module

48.4.1.1.1.318 fobi.contrib.plugins.form_elements.content.content_image_url.fobi_form_elements module

class fobi.contrib.plugins.form_elements.content.content_image_url.fobi_form_elements.ContentImageURLPlugin

Bases: [fobi.base.FormElementPlugin](#)

Content image plugin.

form

alias of [fobi.contrib.plugins.form_elements.content.content_image_url.forms.ContentImageURLForm](#)

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)

Get form field instances.

get_raw_data()

Get raw data.

Might be used in integration plugins.

get_rendered_image()

Get rendered image.

group = 'Content'

name = 'Content image URL'

post_processor()

Post process data.

Always the same.

uid = 'content_image_url'

48.4.1.1.1.319 fobi.contrib.plugins.form_elements.content.content_image_url.forms module

class fobi.contrib.plugins.form_elements.content.content_image_url.forms.ContentImageURLForm(data=None,

files=None,

auto_id='id')

pre-

fix=None,

ini-

tial=None,

er-

ror_class=<class

'django.forms

la-

bel_suffix=None

empty_permi

field_order=1

use_required

ren-

derer=None)

Bases: [django.forms.forms.Form](#), [fobi.base.BasePluginForm](#)

Form for ContentImageURLPlugin.

```
base_fields = {'alt': <django.forms.fields.CharField object>, 'fit_method':
<django.forms.fields.ChoiceField object>, 'size': <django.forms.fields.ChoiceField
object>, 'url': <django.forms.fields.URLField object>}
```

```
declared_fields = {'alt': <django.forms.fields.CharField object>, 'fit_method':
<django.forms.fields.ChoiceField object>, 'size': <django.forms.fields.ChoiceField
object>, 'url': <django.forms.fields.URLField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('url', ''), ('alt', ''), ('fit_method', 'fit_width'),
('size', '500x500')]
```

48.4.1.1.1.320 fobi.contrib.plugins.form_elements.content.content_image_url.settings module

- DEFAULT_SIZE (string)
- SIZE_100x100 (string)
- SIZE_200x200 (string)
- SIZE_300x300 (string)
- SIZE_400x400 (string)
- SIZE_500x500 (string)
- SIZE_600x600 (tuple)
- SIZES (list)

48.4.1.1.1.321 Module contents

48.4.1.1.1.322 fobi.contrib.plugins.form_elements.content.content_markdown package

48.4.1.1.1.323 Submodules

48.4.1.1.1.324 fobi.contrib.plugins.form_elements.content.content_markdown.apps module

```
class fobi.contrib.plugins.form_elements.content.content_markdown.apps.Config(app_name,
                                                                              app_module)
```

Bases: django.apps.config AppConfig

Config.

```
label = 'fobi_contrib_plugins_form_elements_content_content_markdown'
```

```
name = 'fobi.contrib.plugins.form_elements.content.content_markdown'
```

48.4.1.1.1.325 fobi.contrib.plugins.form_elements.content.content_markdown.base module

class `fobi.contrib.plugins.form_elements.content.content_markdown.base.ContentMarkdownPlugin`(*user=None*)

Bases: `fobi.base.FormElementPlugin`

Content markdown plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_markdown.forms.ContentMarkdownForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get the instances of form fields, that plugin contains.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry** (`fobi.models.FormEntry`) –
- **form_element_entries** (`django.db.models.QuerySet`) – Queryset of `fobi.models.FormElementEntry` instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_raw_data()

get_rendered_text()

Get rendered text.

group = 'Content'

html_classes = ['content-markdown']

name = 'Content markdown'

post_processor()

Post-processor (self).

Redefine in your subclassed plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (`django.http.HttpRequest`) is available (`self.request`).

uid = 'content_markdown'

48.4.1.1.1.326 fobi.contrib.plugins.form_elements.content.content_markdown.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.content.content_markdown.fobi_form_elements.ContentMarkdownPlugin`

Bases: `fobi.base.FormElementPlugin`

Content markdown plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_markdown.forms.ContentMarkdownForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get the instances of form fields, that plugin contains.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry** (`fobi.models.FormEntry`) –
- **form_element_entries** (`django.db.models.QuerySet`) – Queryset of `fobi.models.FormElementEntry` instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_raw_data()

get_rendered_text()

Get rendered text.

group = 'Content'

html_classes = ['content-markdown']

name = 'Content markdown'

post_processor()

Post-processor (self).

Redefine in your subclass plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (`django.http.HttpRequest`) is available (`self.request`).

uid = 'content_markdown'

48.4.1.1.1.327 fobi.contrib.plugins.form_elements.content.content_markdown.forms module

```
class fobi.contrib.plugins.form_elements.content.content_markdown.forms.ContentMarkdownForm(data=None,
files=None,
auto_id='id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class 'django.forms.base.forms.ErrorRenderer'>,
'django.forms.base.forms.renderers.html_renderer',
la-
bel_suffix=None,
empty_permitted=True,
field_order=None,
use_required_attribute=True,
ren-
derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

ContentMarkDownForm.

base_fields = {'text': <django.forms.fields.CharField object>}

declared_fields = {'text': <django.forms.fields.CharField object>}

property media

Return all media required to render the widgets on this form.

plugin_data_fields = [('text', '')]

48.4.1.1.1.328 fobi.contrib.plugins.form_elements.content.content_markdown.widgets module

```
class fobi.contrib.plugins.form_elements.content.content_markdown.widgets.BaseContentMarkdownPluginWidget
Bases: fobi.base.FormElementPluginWidget
```

Base content markdown form element plugin widget.

html_classes = ['content-markdown']

plugin_uid = 'content_markdown'

48.4.1.1.1.329 Module contents

48.4.1.1.1.330 fobi.contrib.plugins.form_elements.content.content_richtext package

48.4.1.1.1.331 Submodules

48.4.1.1.1.332 fobi.contrib.plugins.form_elements.content.content_richtext.apps module

```
class fobi.contrib.plugins.form_elements.content.content_richtext.apps.Config(app_name,
app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_content_content_richtext'
name = 'fobi.contrib.plugins.form_elements.content.content_richtext'
```

48.4.1.1.1.333 fobi.contrib.plugins.form_elements.content.content_richtext.base module

class `fobi.contrib.plugins.form_elements.content.content_richtext.base.ContentRichTextPlugin`(*user=None*)
 Bases: `fobi.base.FormElementPlugin`

Content rich text plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_richtext.forms.ContentRichTextForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
 Get the instances of form fields, that plugin contains.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry** (`fobi.models.FormEntry`) –
- **form_element_entries** (`django.db.models.QuerySet`) – Queryset of `fobi.models.FormElementEntry` instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_raw_data()

get_rendered_text()

Get rendered text.

group = 'Content'

name = 'Content rich text'

post_processor()

Post-processor (self).

Redefine in your subclassed plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (`django.http.HttpRequest`) is available (`self.request`).

uid = 'content_richtext'

48.4.1.1.1.334 fobi.contrib.plugins.form_elements.content.content_richtext.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.content.content_richtext.fobi_form_elements.ContentRichTextPlugin`

Bases: `fobi.base.FormElementPlugin`

Content rich text plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_richtext.forms.ContentRichTextForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get the instances of form fields, that plugin contains.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry** (`fobi.models.FormEntry`) –
- **form_element_entries** (`django.db.models.QuerySet`) – Queryset of `fobi.models.FormElementEntry` instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_raw_data()

get_rendered_text()

Get rendered text.

group = 'Content'

name = 'Content rich text'

post_processor()

Post-processor (self).

Redefine in your subclassed plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (`django.http.HttpRequest`) is available (`self.request`).

uid = 'content_richtext'

48.4.1.1.1.335 fobi.contrib.plugins.form_elements.content.content_richtext.forms module

```
class fobi.contrib.plugins.form_elements.content.content_richtext.forms.ContentRichTextForm(data=None,
files=None,
auto_id='id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class 'django.forms.base.forms.ErrorList'>,
'django.forms.base.forms.ErrorList',
la-
bel_suffix=None,
empty_permitted=True,
field_order=None,
use_required_attribute=True,
ren-
derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

`ContentRichTextForm`.

`base_fields` = {'text': <django.forms.fields.CharField object>}

`clean_text()`

`declared_fields` = {'text': <django.forms.fields.CharField object>}

`property media`

Return all media required to render the widgets on this form.

`plugin_data_fields` = [('text', '')]

48.4.1.1.1.336 fobi.contrib.plugins.form_elements.content.content_richtext.widgets module

```
class fobi.contrib.plugins.form_elements.content.content_richtext.widgets.BaseContentRichTextPluginWidget
```

Bases: `fobi.base.FormElementPluginWidget`

Base content rich text form element plugin widget.

`html_classes` = ['content-richtext']

`media_js` = ['ckeditor/ckeditor-init.js', 'ckeditor/ckeditor/ckeditor.js']

`plugin_uid` = 'content_richtext'

48.4.1.1.1.337 Module contents

48.4.1.1.1.338 fobi.contrib.plugins.form_elements.content.content_text package

48.4.1.1.1.339 Submodules

48.4.1.1.1.340 fobi.contrib.plugins.form_elements.content.content_text.apps module

```
class fobi.contrib.plugins.form_elements.content.content_text.apps.Config(app_name,
                                                                    app_module)
```

Bases: `django.apps.config AppConfig`

Config.

label = 'fobi_contrib_plugins_form_elements_content_content_text'

name = 'fobi.contrib.plugins.form_elements.content.content_text'

48.4.1.1.1.341 fobi.contrib.plugins.form_elements.content.content_text.base module

```
class fobi.contrib.plugins.form_elements.content.content_text.base.ContentTextPlugin(user=None)
```

Bases: `fobi.base.FormElementPlugin`

Content text plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

get_raw_data()

Get raw data.

Might be used in integration plugins.

get_rendered_text()

Get rendered text.

group = 'Content'

name = 'Content text'

post_processor()

Post process data.

Always the same.

uid = 'content_text'

48.4.1.1.1.342 fobi.contrib.plugins.form_elements.content.content_text.conf module

`fobi.contrib.plugins.form_elements.content.content_text.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.content.content_text.conf` module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.343 fobi.contrib.plugins.form_elements.content.content_text.defaults module

48.4.1.1.1.344 fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements module

`class fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin(user=)`
 Bases: `fobi.base.FormElementPlugin`

Content text plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
 Get form field instances.

get_raw_data()
 Get raw data.

Might be used in integration plugins.

get_rendered_text()
 Get rendered text.

group = 'Content'

name = 'Content text'

post_processor()
 Post process data.

Always the same.

uid = 'content_text'

48.4.1.1.1.345 fobi.contrib.plugins.form_elements.content.content_text.forms module

```
class fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm(data=None,
                                          files=None,
                                          auto_id='id_%s',
                                          pre-
                                          fix=None,
                                          ini-
                                          tial=None,
                                          er-
                                          ror_class=<class
                                          'django.forms.utils.Error
                                          la-
                                          bel_suffix=None,
                                          empty_permitted=False,
                                          field_order=None,
                                          use_required_attribute=
                                          ren-
                                          derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

Form for ContentTextPlugin.

base_fields = {'text': <django.forms.fields.CharField object>}

clean_text()

Clean text value.

declared_fields = {'text': <django.forms.fields.CharField object>}

property media

Return all media required to render the widgets on this form.

plugin_data_fields = [('text', '')]

48.4.1.1.1.346 fobi.contrib.plugins.form_elements.content.content_text.settings module

48.4.1.1.1.347 Module contents

48.4.1.1.1.348 fobi.contrib.plugins.form_elements.content.content_video package

48.4.1.1.1.349 Submodules

48.4.1.1.1.350 fobi.contrib.plugins.form_elements.content.content_video.apps module

```
class fobi.contrib.plugins.form_elements.content.content_video.apps.Config(app_name,
                                                                           app_module)
```

Bases: `django.apps.config AppConfig`

Config.

label = 'fobi_contrib_plugins_form_elements_content_content_video'

name = 'fobi.contrib.plugins.form_elements.content.content_video'

48.4.1.1.1.351 fobi.contrib.plugins.form_elements.content.content_video.base module

class `fobi.contrib.plugins.form_elements.content.content_video.base.ContentVideoPlugin`(*user=None*)

Bases: `fobi.base.FormElementPlugin`

Content video plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

get_raw_data()

Get raw data.

Might be used in integration plugins.

get_rendered_video()

Get rendered video.

Might be used in integration plugins.

group = 'Content'

name = 'Content video'

post_processor()

Process plugin data.

Always the same.

uid = 'content_video'

48.4.1.1.1.352 fobi.contrib.plugins.form_elements.content.content_video.conf module

`fobi.contrib.plugins.form_elements.content.content_video.conf.get_setting`(*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.content.content_video` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.353 fobi.contrib.plugins.form_elements.content.content_video.defaults module

48.4.1.1.1.354 fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements module

class fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.**ContentVideoPlugin**(use

Bases: *fobi.base.FormElementPlugin*

Content video plugin.

form

alias of *fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm*

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)

Get form field instances.

get_raw_data()

Get raw data.

Might be used in integration plugins.

get_rendered_video()

Get rendered video.

Might be used in integration plugins.

group = 'Content'

name = 'Content video'

post_processor()

Process plugin data.

Always the same.

uid = 'content_video'

48.4.1.1.1.355 fobi.contrib.plugins.form_elements.content.content_video.forms module

class fobi.contrib.plugins.form_elements.content.content_video.forms.**ContentVideoForm**(data=None,

files=None,
auto_id='id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class
'django.forms.utils.Error-
la-
bel_suffix=None,
empty_permitted=False,
field_order=None,
use_required_attribute
ren-
derer=None)

Bases: django.forms.forms.Form, *fobi.base.BasePluginForm*

Form for ContentVideoPlugin.

```
base_fields = {'size': <django.forms.fields.ChoiceField object>, 'title':
<django.forms.fields.CharField object>, 'url': <django.forms.fields.CharField
object>}
```

```
declared_fields = {'size': <django.forms.fields.ChoiceField object>, 'title':
<django.forms.fields.CharField object>, 'url': <django.forms.fields.CharField
object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('title', ''), ('url', ''), ('size', '500x400')]
```

48.4.1.1.1.356 fobi.contrib.plugins.form_elements.content.content_video.settings module

48.4.1.1.1.357 Module contents

48.4.1.1.1.358 Module contents

48.4.1.1.1.359 fobi.contrib.plugins.form_elements.fields package

48.4.1.1.1.360 Subpackages

48.4.1.1.1.361 fobi.contrib.plugins.form_elements.fields.boolean package

48.4.1.1.1.362 Submodules

48.4.1.1.1.363 fobi.contrib.plugins.form_elements.fields.boolean.apps module

```
class fobi.contrib.plugins.form_elements.fields.boolean.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_boolean'
    name = 'fobi.contrib.plugins.form_elements.fields.boolean'
```

48.4.1.1.1.364 fobi.contrib.plugins.form_elements.fields.boolean.base module

```
class fobi.contrib.plugins.form_elements.fields.boolean.base.BooleanSelectPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Boolean select plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.boolean.forms.BooleanSelectForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Boolean'
```

```
uid = 'boolean'
```

48.4.1.1.1.365 fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.BooleanSelectPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Boolean select plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.boolean.forms.BooleanSelectForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Boolean'
    uid = 'boolean'
```

48.4.1.1.1.366 fobi.contrib.plugins.form_elements.fields.boolean.forms module

```
class fobi.contrib.plugins.form_elements.fields.boolean.forms.BooleanSelectForm(data=None,
                                                                                files=None,
                                                                                auto_id='id_%s',
                                                                                pre-
                                                                                fix=None,
                                                                                ini-
                                                                                tial=None,
                                                                                er-
                                                                                ror_class=<class
                                                                                'django.forms.utils.ErrorList'>,
                                                                                la-
                                                                                bel_suffix=None,
                                                                                empty_permitted=False,
                                                                                field_order=None,
                                                                                use_required_attribute=None,
                                                                                ren-
                                                                                derer=None)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm

    Form for BooleanSelectPlugin.

    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
    <django.forms.fields.BooleanField object>, 'label': <django.forms.fields.CharField
    object>, 'name': <django.forms.fields.CharField object>, 'required':
    <django.forms.fields.BooleanField object>}

    declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
    <django.forms.fields.BooleanField object>, 'label': <django.forms.fields.CharField
    object>, 'name': <django.forms.fields.CharField object>, 'required':
    <django.forms.fields.BooleanField object>}

    property media
        Return all media required to render the widgets on this form.
```

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('required', False)]
```

48.4.1.1.1.367 Module contents

48.4.1.1.1.368 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple package

48.4.1.1.1.369 Submodules

48.4.1.1.1.370 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps module

```
class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps.Config(app_name,
                                                                                   app_module)

    Bases: django.apps.config AppConfig

    Config.

    label = 'fobi_contrib_plugins_form_elements_fields_checkbox_select_multiple'

    name = 'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple'
```

48.4.1.1.1.371 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.base module

```
class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.base.CheckboxSelectMultipleInputFormPlugin:
    Bases: fobi.base.FormFieldPlugin

    Checkbox select multiple field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleInputForm

    get_choices()
        Get choices.

        Might be used in integration plugins.

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'

    name = 'Checkbox select multiple'

    prepare_plugin_form_data(cleaned_data)
        Prepare plugin form data.

        Might be used in integration plugins.

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –
```

```
uid = 'checkbox_select_multiple'
```

48.4.1.1.1.372 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf module

```
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf.get_setting(setting,
                                                                                    override=None)
```

Get setting.

Get a setting from *fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.373 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.defaults module

48.4.1.1.1.374 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements.CheckboxSel
Bases: fobi.base.FormFieldPlugin
```

Checkbox select multiple field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleInputForm*

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)

Get form field instances.

```
group = 'Fields'
```

```
name = 'Checkbox select multiple'
```

prepare_plugin_form_data(cleaned_data)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.

- `request` (`django.http.HttpRequest`) –
- `form` (`django.forms.Form`) –

```
uid = 'checkbox_select_multiple'
```

48.4.1.1.1.375 `fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms` module

```
class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleIn
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for `CheckboxSelectMultipleInputPlugin`.

```
base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
clean_initial()
```

Validating the initial value.

```
declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
property media
```

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text',
''), ('initial', ''), ('required', False)]
```

48.4.1.1.1.376 `fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.settings` module

48.4.1.1.1.377 Module contents

48.4.1.1.1.378 `fobi.contrib.plugins.form_elements.fields.date` package

48.4.1.1.1.379 Submodules

48.4.1.1.1.380 `fobi.contrib.plugins.form_elements.fields.date.apps` module

```
class fobi.contrib.plugins.form_elements.fields.date.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_date'
    name = 'fobi.contrib.plugins.form_elements.fields.date'
```

48.4.1.1.1.381 `fobi.contrib.plugins.form_elements.fields.date.base` module

```
class fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin(user=None)
    Bases: fobi.base.ModelFormPlugin
    Date field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Date'
    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.
    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –
    uid = 'date'
```

48.4.1.1.1.382 fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.DateInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Date field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'

    name = 'Date'

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

        Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'date'
```

48.4.1.1.1.383 fobi.contrib.plugins.form_elements.fields.date.forms module

```
class fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           prefix=None,
                                                                           initial=None,
                                                                           error_class=<class
                                                                           'django.forms.utils.ErrorList'>,
                                                                           label_suffix=None,
                                                                           empty_permitted=False,
                                                                           field_order=None,
                                                                           use_required_attribute=None,
                                                                           renderer=None)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm

    Form for DateInputPlugin.

    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
    <django.forms.fields.DateField object>, 'input_formats':
    <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
    object>, 'name': <django.forms.fields.CharField object>, 'required':
    <django.forms.fields.BooleanField object>}

    clean_initial()
        Clean the initial value.
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':  
<django.forms.fields.DateField object>, 'input_formats':  
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField  
object>, 'name': <django.forms.fields.CharField object>, 'required':  
<django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',  
''), ('input_formats', ''), ('required', False)]
```

48.4.1.1.1.384 fobi.contrib.plugins.form_elements.fields.date.widgets module

```
class fobi.contrib.plugins.form_elements.fields.date.widgets.BaseDatePluginWidget(plugin)  
    Bases: fobi.base.FormElementPluginWidget
```

Base date form element plugin widget.

```
plugin_uid = 'date'
```

48.4.1.1.1.385 Module contents**48.4.1.1.1.386 fobi.contrib.plugins.form_elements.fields.date_drop_down package****48.4.1.1.1.387 Submodules****48.4.1.1.1.388 fobi.contrib.plugins.form_elements.fields.date_drop_down.apps module**

```
class fobi.contrib.plugins.form_elements.fields.date_drop_down.apps.Config(app_name,  
                                                                           app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_date_drop_down'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.date_drop_down'
```

48.4.1.1.1.389 fobi.contrib.plugins.form_elements.fields.date_drop_down.base module

```
class fobi.contrib.plugins.form_elements.fields.date_drop_down.base.DateDropDownInputPlugin(user=None)  
    Bases: fobi.base.FormFieldPlugin
```

Date drop down field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.
DateDropDownInputForm*

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)  
    Get form field instances.
```

```
group = 'Fields'
```

```
name = 'Date drop down'
```


submit_plugin_form_data(*form_entry*, *request*, *form*, *form_element_entries*=None, ***kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'date_drop_down'

48.4.1.1.1.390 *fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements* module

class *fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.DateDropDownInputPlug*

Bases: *fobi.base.FormFieldPlugin*

Date drop down field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateDropDownInputForm*

get_form_field_instances(*request*=None, *form_entry*=None, *form_element_entries*=None, ***kwargs*)
Get form field instances.

group = 'Fields'

name = 'Date drop down'

submit_plugin_form_data(*form_entry*, *request*, *form*, *form_element_entries*=None, ***kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'date_drop_down'

48.4.1.1.1.391 fobi.contrib.plugins.form_elements.fields.date_drop_down.forms module

```
class fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateDropDownInputForm(data=None,
files=None,
auto_id='id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class 'django.forms.utils.ErrorList'>,
la-
bel_suffix=None,
empty_permitted=True,
field_order=None,
use_required_attribute=True,
renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for DateDropDownInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'input_formats':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>, 'year_max':
<django.forms.fields.IntegerField object>, 'year_min':
<django.forms.fields.IntegerField object>}
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'input_formats':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>, 'year_max':
<django.forms.fields.IntegerField object>, 'year_min':
<django.forms.fields.IntegerField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('year_min',
''), ('year_max', ''), ('initial', ''), ('input_formats', ''), ('required', False)]
```

48.4.1.1.1.392 Module contents

48.4.1.1.1.393 fobi.contrib.plugins.form_elements.fields.datetime package

48.4.1.1.1.394 Submodules

48.4.1.1.1.395 fobi.contrib.plugins.form_elements.fields.datetime.apps module

```
class fobi.contrib.plugins.form_elements.fields.datetime.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_datetime'
    name = 'fobi.contrib.plugins.form_elements.fields.datetime'
```

48.4.1.1.1.396 fobi.contrib.plugins.form_elements.fields.datetime.base module

```
class fobi.contrib.plugins.form_elements.fields.datetime.base.DateTimeInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    DateTime field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'DateTime'
    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.
        Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –
    uid = 'datetime'
```

48.4.1.1.1.397 fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.DateTimeInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    DateTime field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
```

```
group = 'Fields'
name = 'DateTime'
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

```
uid = 'datetime'
```

48.4.1.1.1.398 `fobi.contrib.plugins.form_elements.fields.datetime.forms` module

```
class fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldInputForm(data=None,
                                                                                      files=None,
                                                                                      auto_id='id_%s',
                                                                                      pre-
                                                                                      fix=None,
                                                                                      ini-
                                                                                      tial=None,
                                                                                      er-
                                                                                      ror_class=<class
                                                                                      'django.forms.utils.ErrorList'
                                                                                      la-
                                                                                      bel_suffix=None,
                                                                                      empty_permitted=False,
                                                                                      field_order=None,
                                                                                      use_required_attribute=None,
                                                                                      ren-
                                                                                      derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for `DateTimeInputPlugin`.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.DateTimeField object>, 'input_formats':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
clean_initial()
```

Clean the initial value.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.DateTimeField object>, 'input_formats':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
property media
```

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('input_formats', ''), ('required', False)]
```

48.4.1.1.1.399 fobi.contrib.plugins.form_elements.fields.datetime.widgets module

```
class fobi.contrib.plugins.form_elements.fields.datetime.widgets.BaseDateTimePluginWidget(plugin)
    Bases: fobi.base.FormElementPluginWidget
    Base datetime form element plugin widget.
    plugin_uid = 'datetime'
```

48.4.1.1.1.400 Module contents

48.4.1.1.1.401 fobi.contrib.plugins.form_elements.fields.decimal package

48.4.1.1.1.402 Submodules

48.4.1.1.1.403 fobi.contrib.plugins.form_elements.fields.decimal.apps module

```
class fobi.contrib.plugins.form_elements.fields.decimal.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_decimal'
    name = 'fobi.contrib.plugins.form_elements.fields.decimal'
```

48.4.1.1.1.404 fobi.contrib.plugins.form_elements.fields.decimal.base module

```
class fobi.contrib.plugins.form_elements.fields.decimal.base.DecimalInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Decimal input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Decimal'
    uid = 'decimal'
```

48.4.1.1.1.405 fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.DecimalInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Decimal input plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Decimal'
    uid = 'decimal'
```

48.4.1.1.1.406 fobi.contrib.plugins.form_elements.fields.decimal.forms module

```
class fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm(data=None,
                                                                                files=None,
                                                                                auto_id='id_%s',
                                                                                prefix=None,
                                                                                initial=None,
                                                                                er-
                                                                                ror_class=<class
                                                                                'django.forms.utils.ErrorList'>,
                                                                                la-
                                                                                bel_suffix=None,
                                                                                empty_permitted=False,
                                                                                field_order=None,
                                                                                use_required_attribute=None,
                                                                                ren-
                                                                                derer=None)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
    Form for DecimalInputPlugin.

    base_fields = {'decimal_places': <django.forms.fields.IntegerField object>,
                  'help_text': <django.forms.fields.CharField object>, 'initial':
                  <django.forms.fields.DecimalField object>, 'label': <django.forms.fields.CharField
                  object>, 'max_digits': <django.forms.fields.IntegerField object>, 'max_value':
                  <django.forms.fields.DecimalField object>, 'min_value':
                  <django.forms.fields.DecimalField object>, 'name': <django.forms.fields.CharField
                  object>, 'placeholder': <django.forms.fields.CharField object>, 'required':
                  <django.forms.fields.BooleanField object>}

    clean()
        Validating the values.
```

```

declared_fields = {'decimal_places': <django.forms.fields.IntegerField object>,
'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.DecimalField object>, 'label': <django.forms.fields.CharField
object>, 'max_digits': <django.forms.fields.IntegerField object>, 'max_value':
<django.forms.fields.DecimalField object>, 'min_value':
<django.forms.fields.DecimalField object>, 'name': <django.forms.fields.CharField
object>, 'placeholder': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}

```

property media

Return all media required to render the widgets on this form.

```

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('max_digits', ''), ('decimal_places', ''), ('min_value', None), ('max_value',
None), ('required', False), ('placeholder', '')]

```

quantize(*value*, *decimal_places*, *max_digits*)

Quantize the decimal value to the configured precision.

48.4.1.1.1.407 Module contents

48.4.1.1.1.408 fobi.contrib.plugins.form_elements.fields.duration package

48.4.1.1.1.409 Submodules

48.4.1.1.1.410 fobi.contrib.plugins.form_elements.fields.duration.apps module

```

class fobi.contrib.plugins.form_elements.fields.duration.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig

```

Config.

label = 'fobi_contrib_plugins_form_elements_fields_duration'

name = 'fobi.contrib.plugins.form_elements.fields.duration'

48.4.1.1.1.411 fobi.contrib.plugins.form_elements.fields.duration.base module

```

class fobi.contrib.plugins.form_elements.fields.duration.base.DurationInputPlugin(user=None)
    Bases: fobi.base.ModelFormPlugin

```

Duration field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.duration.forms.DurationInputForm*

```

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
    Get form field instances.

```

group = 'Fields'

name = 'Duration'

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry*, *request*, *form*, *form_element_entries*=None, ***kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'duration'

48.4.1.1.1.412 fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements module

class *fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements.DurationInputPlugin*(*user=None*)
Bases: *fobi.base.FormFieldPlugin*

Duration field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.duration.forms.DurationInputForm*

get_form_field_instances(*request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)
Get form field instances.

group = 'Fields'

name = 'Duration'

prepare_plugin_form_data(*cleaned_data*)
Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry*, *request*, *form*, *form_element_entries*=None, ***kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'duration'

48.4.1.1.1.413 fobi.contrib.plugins.form_elements.fields.duration.forms module

```
class fobi.contrib.plugins.form_elements.fields.duration.forms.DurationInputForm(data=None,
                                       files=None,
                                       auto_id='id_%s',
                                       pre-
                                       fix=None,
                                       ini-
                                       tial=None,
                                       er-
                                       ror_class=<class
                                       'django.forms.utils.ErrorList'
                                       la-
                                       bel_suffix=None,
                                       empty_permitted=False,
                                       field_order=None,
                                       use_required_attribute=None,
                                       ren-
                                       derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for DurationInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.DurationField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

`clean()`

Validation.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.DurationField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property `media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('required', False), ('weeks', ''), ('days', ''), ('hours', ''), ('minutes',
''), ('seconds', ''), ('placeholder', '')]
```

48.4.1.1.1.414 Module contents

48.4.1.1.1.415 fobi.contrib.plugins.form_elements.fields.email package

48.4.1.1.1.416 Submodules

48.4.1.1.1.417 fobi.contrib.plugins.form_elements.fields.email.apps module

```
class fobi.contrib.plugins.form_elements.fields.email.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_email'
    name = 'fobi.contrib.plugins.form_elements.fields.email'
```

48.4.1.1.1.418 fobi.contrib.plugins.form_elements.fields.email.base module

```
class fobi.contrib.plugins.form_elements.fields.email.base.EmailInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Email input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Email'
    uid = 'email'
```

48.4.1.1.1.419 fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements.EmailInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Email input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Email'
    uid = 'email'
```

48.4.1.1.1.420 fobi.contrib.plugins.form_elements.fields.email.forms module

```
class fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm(data=None,
                                  files=None,
                                  auto_id='id_%s',
                                  prefix=None,
                                  initial=None, error_class=<class
                                  'django.forms.utils.ErrorList'>,
                                  label_suffix=None,
                                  empty_permitted=False,
                                  field_order=None,
                                  use_required_attribute=None,
                                  renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for EmailInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.EmailField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

`clean()`

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.EmailField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

`property media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('max_length', '255'), ('required', False), ('placeholder', '')]
```

48.4.1.1.1.421 Module contents

48.4.1.1.1.422 fobi.contrib.plugins.form_elements.fields.file package

48.4.1.1.1.423 Submodules

48.4.1.1.1.424 fobi.contrib.plugins.form_elements.fields.file.apps module

```
class fobi.contrib.plugins.form_elements.fields.file.apps.Config(app_name, app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_file'
name = 'fobi.contrib.plugins.form_elements.fields.file'
```

48.4.1.1.1.425 fobi.contrib.plugins.form_elements.fields.file.base module

class `fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin`(*user=None*)

Bases: `fobi.base.FormFieldPlugin`

File field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

name = 'File'

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Parameters *cleaned_data* –

Returns

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process file upload.

Handling the posted data for file plugin when form is submitted. This method affects the original form and that's why it returns it.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** –

uid = 'file'

48.4.1.1.1.426 fobi.contrib.plugins.form_elements.fields.file.conf module

`fobi.contrib.plugins.form_elements.fields.file.conf.get_setting`(*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.conf` module, falling back to the default.

If *override* is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

48.4.1.1.1.427 fobi.contrib.plugins.form_elements.fields.file.defaults module

48.4.1.1.1.428 fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.FileInputPlugin`(*user=None*)

Bases: `fobi.base.FormFieldPlugin`

File field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

name = 'File'

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Parameters *cleaned_data* –

Returns

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process file upload.

Handling the posted data for file plugin when form is submitted. This method affects the original form and that's why it returns it.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** –

uid = 'file'

48.4.1.1.1.429 fobi.contrib.plugins.form_elements.fields.file.forms module

```
class fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm(data=None,
                                files=None,
                                auto_id='id_%s',
                                prefix=None,
                                initial=None,
                                error_class=<class
                                'django.forms.utils.ErrorList'>,
                                label_suffix=None,
                                empty_permitted=False,
                                field_order=None,
                                use_required_attribute=None,
                                renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for FileInputPlugin.

```
base_fields = {'allowed_extensions': <django.forms.fields.CharField object>,
               'help_text': <django.forms.fields.CharField object>, 'initial':
               <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
               object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
               <django.forms.fields.CharField object>, 'required':
               <django.forms.fields.BooleanField object>}
```

clean()

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'allowed_extensions': <django.forms.fields.CharField object>,
                  'help_text': <django.forms.fields.CharField object>, 'initial':
                  <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
                  object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
                  <django.forms.fields.CharField object>, 'required':
                  <django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
'', ('max_length', '255'), ('required', False), ('allowed_extensions', ''))]
```

48.4.1.1.1.430 fobi.contrib.plugins.form_elements.fields.file.settings module

- FILES_UPLOAD_DIR (string)

48.4.1.1.1.431 Module contents

48.4.1.1.1.432 fobi.contrib.plugins.form_elements.fields.float package

48.4.1.1.1.433 Submodules

48.4.1.1.1.434 fobi.contrib.plugins.form_elements.fields.float.apps module

```
class fobi.contrib.plugins.form_elements.fields.float.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_float'
    name = 'fobi.contrib.plugins.form_elements.fields.float'
```

48.4.1.1.1.435 fobi.contrib.plugins.form_elements.fields.float.base module

```
class fobi.contrib.plugins.form_elements.fields.float.base.FloatInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Float input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Float'
    uid = 'float'
```

48.4.1.1.1.436 fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements.FloatInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Float input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Float'
```

```
uid = 'float'
```

48.4.1.1.1.437 fobi.contrib.plugins.form_elements.fields.float.forms module

```
class fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm(data=None,
                                   files=None,
                                   auto_id='id_%s',
                                   prefix=None,
                                   initial=None, error_class=<class
                                   'django.forms.utils.ErrorList'>,
                                   label_suffix=None,
                                   empty_permitted=False,
                                   field_order=None,
                                   use_required_attribute=None,
                                   renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for FloatInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.FloatField object>, 'label': <django.forms.fields.CharField
object>, 'max_value': <django.forms.fields.FloatField object>, 'min_value':
<django.forms.fields.FloatField object>, 'name': <django.forms.fields.CharField
object>, 'placeholder': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
clean()
```

Validating the values.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.FloatField object>, 'label': <django.forms.fields.CharField
object>, 'max_value': <django.forms.fields.FloatField object>, 'min_value':
<django.forms.fields.FloatField object>, 'name': <django.forms.fields.CharField
object>, 'placeholder': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
property media
```

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('min_value', None), ('max_value', None), ('required', False), ('placeholder',
'')]
```


48.4.1.1.1.438 Module contents

48.4.1.1.1.439 fobi.contrib.plugins.form_elements.fields.hidden package

48.4.1.1.1.440 Submodules

48.4.1.1.1.441 fobi.contrib.plugins.form_elements.fields.hidden.apps module

```
class fobi.contrib.plugins.form_elements.fields.hidden.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_hidden'
    name = 'fobi.contrib.plugins.form_elements.fields.hidden'
```

48.4.1.1.1.442 fobi.contrib.plugins.form_elements.fields.hidden.base module

```
class fobi.contrib.plugins.form_elements.fields.hidden.base.HiddenInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Hidden field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    is_hidden = True
    name = 'Hidden'
    uid = 'hidden'
```

48.4.1.1.1.443 fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements.HiddenInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Hidden field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    is_hidden = True
    name = 'Hidden'
    uid = 'hidden'
```

48.4.1.1.1.444 fobi.contrib.plugins.form_elements.fields.hidden.forms module

```
class fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm(data=None,
                                     files=None,
                                     auto_id='id_%s',
                                     prefix=None,
                                     initial=None,
                                     error_class=<class
                                     'django.forms.utils.ErrorList'>,
                                     label_suffix=None,
                                     empty_permitted=False,
                                     field_order=None,
                                     use_required_attribute=None,
                                     renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for HiddenInputPlugin.

```
base_fields = {'initial': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'max_length':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

clean()

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'initial': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'max_length':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('initial', ''), ('max_length',
'255'), ('required', False)]
```

48.4.1.1.1.445 Module contents

48.4.1.1.1.446 fobi.contrib.plugins.form_elements.fields.input package

48.4.1.1.1.447 Submodules

48.4.1.1.1.448 fobi.contrib.plugins.form_elements.fields.input.apps module

```
class fobi.contrib.plugins.form_elements.fields.input.apps.Config(app_name, app_module)
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_input'
name = 'fobi.contrib.plugins.form_elements.fields.input'
```

48.4.1.1.1.449 fobi.contrib.plugins.form_elements.fields.input.base module

```
class fobi.contrib.plugins.form_elements.fields.input.base.InputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Input field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.input.forms.InputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Input'
    uid = 'input'
```

48.4.1.1.1.450 fobi.contrib.plugins.form_elements.fields.input.constants module

48.4.1.1.1.451 fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements.InputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Input field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.input.forms.InputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Input'
    uid = 'input'
```

48.4.1.1.1.452 fobi.contrib.plugins.form_elements.fields.input.forms module

```
class fobi.contrib.plugins.form_elements.fields.input.forms.InputForm(data=None, files=None,
                                                                    auto_id='id_%s',
                                                                    prefix=None,
                                                                    initial=None,
                                                                    error_class=<class
                                                                    'django.forms.utils.ErrorList'>,
                                                                    label_suffix=None,
                                                                    empty_permitted=False,
                                                                    field_order=None,
                                                                    use_required_attribute=None,
                                                                    renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for InputPlugin.

```
base_fields = {'autocomplete_value': <django.forms.fields.BooleanField object>,
               'autofocus_value': <django.forms.fields.BooleanField object>, 'disabled_value':
               <django.forms.fields.BooleanField object>, 'help_text':
               <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
               object>, 'label': <django.forms.fields.CharField object>, 'list_value':
               <django.forms.fields.CharField object>, 'max_length':
               <django.forms.fields.IntegerField object>, 'max_value':
               <django.forms.fields.CharField object>, 'min_value': <django.forms.fields.CharField
               object>, 'multiple_value': <django.forms.fields.BooleanField object>, 'name':
               <django.forms.fields.CharField object>, 'pattern_value':
               <django.forms.fields.CharField object>, 'placeholder':
               <django.forms.fields.CharField object>, 'readonly_value':
               <django.forms.fields.BooleanField object>, 'required':
               <django.forms.fields.BooleanField object>, 'step_value':
               <django.forms.fields.IntegerField object>, 'type_value':
               <django.forms.fields.ChoiceField object>}
```

`clean()`

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'autocomplete_value': <django.forms.fields.BooleanField object>,
                  'autofocus_value': <django.forms.fields.BooleanField object>, 'disabled_value':
                  <django.forms.fields.BooleanField object>, 'help_text':
                  <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
                  object>, 'label': <django.forms.fields.CharField object>, 'list_value':
                  <django.forms.fields.CharField object>, 'max_length':
                  <django.forms.fields.IntegerField object>, 'max_value':
                  <django.forms.fields.CharField object>, 'min_value': <django.forms.fields.CharField
                  object>, 'multiple_value': <django.forms.fields.BooleanField object>, 'name':
                  <django.forms.fields.CharField object>, 'pattern_value':
                  <django.forms.fields.CharField object>, 'placeholder':
                  <django.forms.fields.CharField object>, 'readonly_value':
                  <django.forms.fields.BooleanField object>, 'required':
                  <django.forms.fields.BooleanField object>, 'step_value':
                  <django.forms.fields.IntegerField object>, 'type_value':
                  <django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
 ''), ('max_length', '255'), ('required', False), ('placeholder', ''),
 ('autocomplete_value', 'off'), ('autofocus_value', False), ('disabled_value',
 False), ('list_value', ''), ('max_value', ''), ('min_value', ''), ('multiple_value',
 False), ('pattern_value', ''), ('readonly_value', False), ('step_value', ''),
 ('type_value', 'text')]
```

48.4.1.1.1.453 Module contents**48.4.1.1.1.454 fobi.contrib.plugins.form_elements.fields.integer package****48.4.1.1.1.455 Submodules****48.4.1.1.1.456 fobi.contrib.plugins.form_elements.fields.integer.apps module**

```
class fobi.contrib.plugins.form_elements.fields.integer.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_integer'
    name = 'fobi.contrib.plugins.form_elements.fields.integer'
```

48.4.1.1.1.457 fobi.contrib.plugins.form_elements.fields.integer.base module

```
class fobi.contrib.plugins.form_elements.fields.integer.base.IntegerInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Integer input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Integer'
    uid = 'integer'
```

48.4.1.1.1.458 fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.IntegerInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Integer input plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Integer'
    uid = 'integer'
```

48.4.1.1.1.459 fobi.contrib.plugins.form_elements.fields.integer.forms module

```
class fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm(data=None,
                                     files=None,
                                     auto_id='id_%s',
                                     prefix=None,
                                     initial=None,
                                     error_class=<class
                                     'django.forms.utils.ErrorList'>,
                                     label_suffix=None,
                                     empty_permitted=False,
                                     field_order=None,
                                     use_required_attribute=None,
                                     renderer=None)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
    Form for IntegerInputPlugin.

    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
    <django.forms.fields.IntegerField object>, 'label': <django.forms.fields.CharField
    object>, 'max_value': <django.forms.fields.IntegerField object>, 'min_value':
    <django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
    object>, 'placeholder': <django.forms.fields.CharField object>, 'required':
    <django.forms.fields.BooleanField object>}

    clean()
        Validating the values.

    declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
    <django.forms.fields.IntegerField object>, 'label': <django.forms.fields.CharField
    object>, 'max_value': <django.forms.fields.IntegerField object>, 'min_value':
    <django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
    object>, 'placeholder': <django.forms.fields.CharField object>, 'required':
    <django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
    ''), ('min_value', None), ('max_value', None), ('required', False), ('placeholder',
    '')]
```

48.4.1.1.1.460 Module contents

48.4.1.1.1.461 fobi.contrib.plugins.form_elements.fields.ip_address package

48.4.1.1.1.462 Submodules

48.4.1.1.1.463 fobi.contrib.plugins.form_elements.fields.ip_address.apps module

```
class fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config(app_name,
    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_ip_address'
    name = 'fobi.contrib.plugins.form_elements.fields.ip_address'
```

48.4.1.1.1.464 fobi.contrib.plugins.form_elements.fields.ip_address.base module

```
class fobi.contrib.plugins.form_elements.fields.ip_address.base.IPAddressInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    IP address field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.ip_address.forms.
        IPAddressInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'IP address'
    uid = 'ip_address'
```

48.4.1.1.1.465 fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.IPAddressInputPlugin(user=
    None)
    Bases: fobi.base.FormFieldPlugin
    IP address field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.ip_address.forms.
        IPAddressInputForm
```

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
```

Get form field instances.

```
group = 'Fields'
```

```
name = 'IP address'
```

```
uid = 'ip_address'
```

48.4.1.1.1.466 fobi.contrib.plugins.form_elements.fields.ip_address.forms module

```
class fobi.contrib.plugins.form_elements.fields.ip_address.forms.IPAddressInputForm(data=None,
                                          files=None,
                                          auto_id='id_%s',
                                          pre-
                                          fix=None,
                                          ini-
                                          tial=None,
                                          er-
                                          ror_class=<class
                                          'django.forms.utils.Error
                                          la-
                                          bel_suffix=None,
                                          empty_permitted=False,
                                          field_order=None,
                                          use_required_attribute=l
                                          ren-
                                          derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for IPAddressInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'protocol':
<django.forms.fields.ChoiceField object>, 'required':
<django.forms.fields.BooleanField object>, 'unpack_ipv4':
<django.forms.fields.BooleanField object>}
```

```
clean()
```

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'protocol':
<django.forms.fields.ChoiceField object>, 'required':
<django.forms.fields.BooleanField object>, 'unpack_ipv4':
<django.forms.fields.BooleanField object>}
```

```
property media
```

Return all media required to render the widgets on this form.


```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('protocol', ''), ('unpack_ipv4', False), ('max_length', '255'), ('required',
False), ('placeholder', '')]
```

48.4.1.1.1.467 Module contents

48.4.1.1.1.468 fobi.contrib.plugins.form_elements.fields.null_boolean package

48.4.1.1.1.469 Submodules

48.4.1.1.1.470 fobi.contrib.plugins.form_elements.fields.null_boolean.apps module

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_null_boolean'
    name = 'fobi.contrib.plugins.form_elements.fields.null_boolean'
```

48.4.1.1.1.471 fobi.contrib.plugins.form_elements.fields.null_boolean.base module

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.base.NullBooleanSelectPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Null boolean select plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.null_boolean.forms.
        NullBooleanSelectForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Null boolean'
    uid = 'null_boolean'
```

48.4.1.1.1.472 fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.NullBooleanSelectPlugin
    Bases: fobi.base.FormFieldPlugin
    Null boolean select plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.null_boolean.forms.
        NullBooleanSelectForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
```

```
group = 'Fields'
name = 'Null boolean'
uid = 'null_boolean'
```

48.4.1.1.1.473 fobi.contrib.plugins.form_elements.fields.null_boolean.forms module

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.forms.NullBooleanSelectForm(data=None,
files=None,
auto_id='id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class
'django.forms.util
la-
bel_suffix=None,
empty_permitted=
field_order=None,
use_required_attri
ren-
derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for NullBooleanSelectPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.NullBooleanField object>, 'label':
<django.forms.fields.CharField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.NullBooleanField object>, 'label':
<django.forms.fields.CharField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('required', False)]
```

48.4.1.1.1.474 Module contents

48.4.1.1.1.475 fobi.contrib.plugins.form_elements.fields.password package

48.4.1.1.1.476 Submodules

48.4.1.1.1.477 fobi.contrib.plugins.form_elements.fields.password.apps module

```
class fobi.contrib.plugins.form_elements.fields.password.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_password'
    name = 'fobi.contrib.plugins.form_elements.fields.password'
```

48.4.1.1.1.478 fobi.contrib.plugins.form_elements.fields.password.base module

```
class fobi.contrib.plugins.form_elements.fields.password.base.PasswordInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Password field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Password'
    uid = 'password'
```

48.4.1.1.1.479 fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements.PasswordInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Password field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Password'
    uid = 'password'
```

48.4.1.1.1.480 fobi.contrib.plugins.form_elements.fields.password.forms module

```
class fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm(data=None,
                                         files=None,
                                         auto_id='id_%s',
                                         pre-
                                         fix=None,
                                         ini-
                                         tial=None,
                                         er-
                                         ror_class=<class
                                         'django.forms.utils.ErrorList'
                                         la-
                                         bel_suffix=None,
                                         empty_permitted=False,
                                         field_order=None,
                                         use_required_attribute=None,
                                         ren-
                                         derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for PasswordInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

clean()

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('max_length', '255'), ('required', False), ('placeholder', '')]
```

48.4.1.1.1.481 Module contents

48.4.1.1.1.482 fobi.contrib.plugins.form_elements.fields.radio package

48.4.1.1.1.483 Submodules

48.4.1.1.1.484 fobi.contrib.plugins.form_elements.fields.radio.apps module

```
class fobi.contrib.plugins.form_elements.fields.radio.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_radio'
    name = 'fobi.contrib.plugins.form_elements.fields.radio'
```

48.4.1.1.1.485 fobi.contrib.plugins.form_elements.fields.radio.base module

```
class fobi.contrib.plugins.form_elements.fields.radio.base.RadioInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Radio field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.radio.forms.RadioInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Radio'

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'radio'
```

48.4.1.1.1.486 fobi.contrib.plugins.form_elements.fields.radio.conf module

```
fobi.contrib.plugins.form_elements.fields.radio.conf.get_setting(setting, override=None)
    Get setting.

    Get a setting from fobi.contrib.plugins.form_elements.fields.radio conf module, falling back to the default.
    If override is not None, it will be used instead of the setting.
```

Parameters

- **setting** – String with setting name

- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.487 fobi.contrib.plugins.form_elements.fields.radio.defaults module

48.4.1.1.1.488 fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements.**RadioInputPlugin**(*user=None*)

Bases: *fobi.base.FormFieldPlugin*

Radio field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.radio.forms.RadioInputForm*

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

name = 'Radio'

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'radio'

48.4.1.1.1.489 fobi.contrib.plugins.form_elements.fields.radio.forms module

class fobi.contrib.plugins.form_elements.fields.radio.forms.**RadioInputForm**(*data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None, renderer=None*)

Bases: *django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm*

Form for RadioInputPlugin.

```
base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}

clean_initial()
    Validating the initial value.

declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}

property media
    Return all media required to render the widgets on this form.

plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text',
''), ('initial', ''), ('required', False)]
```

48.4.1.1.1.490 fobi.contrib.plugins.form_elements.fields.radio.settings module

48.4.1.1.1.491 Module contents

48.4.1.1.1.492 fobi.contrib.plugins.form_elements.fields.range_select package

48.4.1.1.1.493 Submodules

48.4.1.1.1.494 fobi.contrib.plugins.form_elements.fields.range_select.apps module

```
class fobi.contrib.plugins.form_elements.fields.range_select.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_range_select'
    name = 'fobi.contrib.plugins.form_elements.fields.range_select'
```

48.4.1.1.1.495 fobi.contrib.plugins.form_elements.fields.range_select.base module

```
class fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSelectInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Range select input plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.range_select.forms.
RangeSelectInputForm

    get_choices()
        Get choices.

        Might be used in integration plugins.
```

```

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
    Get form field instances.

get_initial()
    Get initial value.

    Might be used in integration plugins.

group = 'Fields'
name = 'Range select'
uid = 'range_select'

```

48.4.1.1.1.496 fobi.contrib.plugins.form_elements.fields.range_select.conf module

```

fobi.contrib.plugins.form_elements.fields.range_select.conf.get_setting(setting,
                                                                    override=None)
    Get setting.

    Get a setting from fobi.contrib.plugins.form_elements.fields.range_select conf module, falling back to the default.

    If override is not None, it will be used instead of the setting.

```

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.497 fobi.contrib.plugins.form_elements.fields.range_select.defaults module

48.4.1.1.1.498 fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements module

```

class fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements.RangeSelectInputPlugin(
    Bases: fobi.base.FormFieldPlugin
    Range select input plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.range_select.forms.RangeSelectInputForm

    get_choices()
        Get choices.

        Might be used in integration plugins.

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    get_initial()
        Get initial value.

        Might be used in integration plugins.

    group = 'Fields'
    name = 'Range select'

```



```
uid = 'range_select'
```

48.4.1.1.1.499 fobi.contrib.plugins.form_elements.fields.range_select.forms module

```
class fobi.contrib.plugins.form_elements.fields.range_select.forms.RangeSelectInputForm(data=None,
                                                                                       files=None,
                                                                                       auto_id='id_%s',
                                                                                       pre-
                                                                                       fix=None,
                                                                                       ini-
                                                                                       tial=None,
                                                                                       er-
                                                                                       ror_class=<class
                                                                                       'django.forms.utils.L
                                                                                       la-
                                                                                       bel_suffix=None,
                                                                                       empty_permitted=F
                                                                                       field_order=None,
                                                                                       use_required_attrib
                                                                                       ren-
                                                                                       derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for RangeSelectInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.IntegerField object>, 'label': <django.forms.fields.CharField
object>, 'max_value': <django.forms.fields.IntegerField object>, 'min_value':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>, 'step':
<django.forms.fields.IntegerField object>}
```

```
clean()
```

Validating the values.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.IntegerField object>, 'label': <django.forms.fields.CharField
object>, 'max_value': <django.forms.fields.IntegerField object>, 'min_value':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>, 'step':
<django.forms.fields.IntegerField object>}
```

```
property media
```

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('min_value', 0), ('max_value',
100), ('step', 1), ('help_text', ''), ('initial', 50), ('required', False)]
```

48.4.1.1.1.500 fobi.contrib.plugins.form_elements.fields.range_select.settings module

48.4.1.1.1.501 Module contents

48.4.1.1.1.502 fobi.contrib.plugins.form_elements.fields.regex package

48.4.1.1.1.503 Submodules

48.4.1.1.1.504 fobi.contrib.plugins.form_elements.fields.regex.apps module

```
class fobi.contrib.plugins.form_elements.fields.regex.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_regex'
    name = 'fobi.contrib.plugins.form_elements.fields.regex'
```

48.4.1.1.1.505 fobi.contrib.plugins.form_elements.fields.regex.base module

```
class fobi.contrib.plugins.form_elements.fields.regex.base.RegexInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Regex field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Regex'
    uid = 'regex'
```

48.4.1.1.1.506 fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.RegexInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Regex field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Regex'
    uid = 'regex'
```

48.4.1.1.1.507 fobi.contrib.plugins.form_elements.fields.regex.forms module

```
class fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm(data=None,
                                                                            files=None,
                                                                            auto_id='id_%s',
                                                                            prefix=None,
                                                                            initial=None, error_class=<class
                                                                            'django.forms.utils.ErrorList'>,
                                                                            label_suffix=None,
                                                                            empty_permitted=False,
                                                                            field_order=None,
                                                                            use_required_attribute=None,
                                                                            renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for RegexInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'regex': <django.forms.fields.RegexField
object>, 'required': <django.forms.fields.BooleanField object>}
```

`clean()`

Validation.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'regex': <django.forms.fields.RegexField
object>, 'required': <django.forms.fields.BooleanField object>}
```

`property media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('regex', ''), ('max_length', '255'), ('required', False), ('placeholder', '')]
```

48.4.1.1.1.508 Module contents

48.4.1.1.1.509 fobi.contrib.plugins.form_elements.fields.select package

48.4.1.1.1.510 Submodules

48.4.1.1.1.511 fobi.contrib.plugins.form_elements.fields.select.apps module

```
class fobi.contrib.plugins.form_elements.fields.select.apps.Config(app_name, app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_select'
name = 'fobi.contrib.plugins.form_elements.fields.select'
```

48.4.1.1.1.512 fobi.contrib.plugins.form_elements.fields.select.base module

```
class fobi.contrib.plugins.form_elements.fields.select.base.SelectInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Select field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm

    get_choices()
        Get choices.

        Might be used in integration plugins.

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Select'

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

        Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'select'
```

48.4.1.1.1.513 fobi.contrib.plugins.form_elements.fields.select.conf module

```
fobi.contrib.plugins.form_elements.fields.select.conf.get_setting(setting, override=None)
    Get setting.

    Get a setting from fobi.contrib.plugins.form_elements.fields.select conf module, falling back to the default.
    If override is not None, it will be used instead of the setting.

    Parameters
    • setting – String with setting name
    • override – Value to use when no setting is available. Defaults to None.

    Returns Setting value.
```

48.4.1.1.1.514 fobi.contrib.plugins.form_elements.fields.select.defaults module

48.4.1.1.1.515 fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.SelectInputPlugin`(*user=None*)

Bases: `fobi.base.FormFieldPlugin`

Select field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm`

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

name = 'Select'

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select'

48.4.1.1.1.516 fobi.contrib.plugins.form_elements.fields.select.forms module

class `fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm`(*data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None, renderer=None*)

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for `SelectInputPlugin`.

```
base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

clean_initial()

Validating the initial value.

```
declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text',
''), ('initial', ''), ('required', False)]
```

48.4.1.1.1.517 fobi.contrib.plugins.form_elements.fields.select.settings module

48.4.1.1.1.518 Module contents

48.4.1.1.1.519 fobi.contrib.plugins.form_elements.fields.select_model_object package

48.4.1.1.1.520 Submodules

48.4.1.1.1.521 fobi.contrib.plugins.form_elements.fields.select_model_object.apps module

```
class fobi.contrib.plugins.form_elements.fields.select_model_object.apps.Config(app_name,
                                                                 app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_select_model_object'
    name = 'fobi.contrib.plugins.form_elements.fields.select_model_object'
```

48.4.1.1.1.522 fobi.contrib.plugins.form_elements.fields.select_model_object.base module

```
class fobi.contrib.plugins.form_elements.fields.select_model_object.base.SelectModelObjectInputPlugin(u
    Bases: fobi.base.FormFieldPlugin
    Select model object field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.select_model_object.forms.
        SelectModelObjectInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
```

get_queryset()

Get queryset.

Might be used in integration packages.

group = 'Fields'

name = 'Select model object'

prepare_plugin_form_data(cleaned_data)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'select_model_object'

48.4.1.1.1.523 fobi.contrib.plugins.form_elements.fields.select_model_object.conf module

fobi.contrib.plugins.form_elements.fields.select_model_object.conf.get_setting(setting, override=None)

Get setting.

Get a setting from *fobi.contrib.plugins.form_elements.fields.select_model_object* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.524 fobi.contrib.plugins.form_elements.fields.select_model_object.defaults module

48.4.1.1.1.525 fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements.SelectModelObject

Bases: *fobi.base.FormFieldPlugin*

Select model object field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectInputForm*

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)

Get form field instances.

get_queryset()

Get queryset.

Might be used in integration packages.

group = 'Fields'

name = 'Select model object'

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry*, *request*, *form*, *form_element_entries*=None, *kwargs*)**

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'select_model_object'

48.4.1.1.1.526 *fobi.contrib.plugins.form_elements.fields.select_model_object.forms* module

class *fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectInputForm*(**args*, ***kwargs*)

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for *SelectModelObjectPlugin*.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'model': <django.forms.fields.ChoiceField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'model': <django.forms.fields.ChoiceField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''),
('initial', ''), ('required', False)]
```


48.4.1.1.1.527 fobi.contrib.plugins.form_elements.fields.select_model_object.settings module

48.4.1.1.1.528 Module contents

48.4.1.1.1.529 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object package

48.4.1.1.1.530 Submodules

48.4.1.1.1.531 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps module

```
class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps.Config(app_name,
                                                                                   app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_select_mptt_model_object'
    name = 'fobi.contrib.plugins.form_elements.fields.select_mptt_model_object'
```

48.4.1.1.1.532 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base module

```
class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base.SelectMPTTModelObjectInputForm
    Bases: fobi.base.FormFieldPlugin
    Select MPTT model object field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Select MPTT model object'

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'select_mptt_model_object'
```

48.4.1.1.1.533 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf module

`fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.534 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults module

48.4.1.1.1.535 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.fobi_form_elements module

`class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.fobi_form_elements.SelectMPTTModelObjectFieldPlugin`

Bases: `fobi.base.FormFieldPlugin`

Select MPTT model object field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectInputForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
Get form field instances.

group = 'Fields'

name = 'Select MPTT model object'

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_mptt_model_object'

48.4.1.1.1.536 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms module

`class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectInp`

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for SelectMPTTModelObjectPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'model': <django.forms.fields.ChoiceField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'model': <django.forms.fields.ChoiceField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property `media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''),
('initial', ''), ('required', False)]
```

48.4.1.1.1.537 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.settings module

48.4.1.1.1.538 Module contents

48.4.1.1.1.539 fobi.contrib.plugins.form_elements.fields.select_multiple package

48.4.1.1.1.540 Submodules

48.4.1.1.1.541 fobi.contrib.plugins.form_elements.fields.select_multiple.apps module

`class fobi.contrib.plugins.form_elements.fields.select_multiple.apps.Config`(*app_name*,
app_module)

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_select_multiple'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.select_multiple'
```

48.4.1.1.1.542 fobi.contrib.plugins.form_elements.fields.select_multiple.base module

class `fobi.contrib.plugins.form_elements.fields.select_multiple.base.SelectMultipleInputPlugin`(*user=None*)

Bases: `fobi.base.FormFieldPlugin`

Select multiple field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleInputForm`

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

name = 'Select multiple'

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_multiple'

48.4.1.1.1.543 fobi.contrib.plugins.form_elements.fields.select_multiple.conf module

`fobi.contrib.plugins.form_elements.fields.select_multiple.conf.get_setting`(*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_multiple` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.544 fobi.contrib.plugins.form_elements.fields.select_multiple.defaults module

48.4.1.1.1.545 fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultipleInputP`

Bases: `fobi.base.FormFieldPlugin`

Select multiple field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleInputForm`

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

name = 'Select multiple'

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_multiple'

48.4.1.1.1.546 fobi.contrib.plugins.form_elements.fields.select_multiple.forms module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleInputForm(data=None,
files=None,
auto_id='id',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<
'django.form
la-
bel_suffix=N
empty_perm
field_order=
use_require
ren-
derer=None
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for `SelectMultipleInputPlugin`.

```
base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

`clean_initial()`

Validating the initial value.

```
declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property `media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text',
''), ('initial', ''), ('required', False)]
```

48.4.1.1.1.547 fobi.contrib.plugins.form_elements.fields.select_multiple.settings module

48.4.1.1.1.548 Module contents

48.4.1.1.1.549 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects package

48.4.1.1.1.550 Submodules

48.4.1.1.1.551 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps.Config(app_name,
                                                                                       app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_select_multiple_model_objects'
    name = 'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects'
```

48.4.1.1.1.552 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.base module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.base.SelectMultipleModelObjectsInputForm
    Bases: fobi.base.ModelFormPlugin
    Select multiple model objects field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.
        forms.SelectMultipleModelObjectsInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    get_queryset()
        Get queryset.

        Might be used in integration packages.

    group = 'Fields'
    name = 'Select multiple model objects'

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'select_multiple_model_objects'
```

48.4.1.1.1.553 `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf` module

`fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.554 `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults` module

48.4.1.1.1.555 `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements` module

`class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements.SelectMultipleModelObjectsFieldPlugin`

Bases: `fobi.base.FormFieldPlugin`

Select multiple model objects field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms.SelectMultipleModelObjectsInputForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
Get form field instances.

get_queryset()

Get queryset.

Might be used in integration packages.

group = 'Fields'

name = 'Select multiple model objects'

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_multiple_model_objects'

48.4.1.1.1.556 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms.SelectMultipleModelObjectsPlugin
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for SelectMultipleModelObjectsPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'model': <django.forms.fields.ChoiceField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'model': <django.forms.fields.ChoiceField object>, 'name':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property `media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''),
('initial', ''), ('required', False)]
```

48.4.1.1.1.557 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.settings module

48.4.1.1.1.558 Module contents

48.4.1.1.1.559 fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects package

48.4.1.1.1.560 Submodules

48.4.1.1.1.561 fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps.Config(app_name,
app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label =
'fobi_contrib_plugins_form_elements_fields_select_multiple_mptt_model_objects'
```

```
name =
'fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects'
```

48.4.1.1.1.562 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.base` module

class `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.base.SelectMultipleM`

Bases: `fobi.base.FormFieldPlugin`

Select multiple MPTT model object field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms.SelectMultipleMPTTModelObjectsInputForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
Get form field instances.

group = 'Fields'

name = 'Select multiple MPTT model objects'

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_multiple_mptt_model_objects'

48.4.1.1.1.563 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf` module

`fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf.get_setting`(*setting, override=None*)

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.564 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults` module

48.4.1.1.1.565 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_elements` module

class `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_elements.S`
 Bases: `fobi.base.FormFieldPlugin`

Select multiple MPTT model object field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms.SelectMultipleMPTTModelObjectsInputForm`

get_form_field_instances(`request=None, form_entry=None, form_element_entries=None, **kwargs`)
 Get form field instances.

group = 'Fields'

name = 'Select multiple MPTT model objects'

submit_plugin_form_data(`form_entry, request, form, form_element_entries=None, **kwargs`)
 Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_multiple_mptt_model_objects'

48.4.1.1.1.566 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms` module

class `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms.SelectMultiple`

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for `SelectMultipleMPTTModelObjectsPlugin`.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField object>, 'model': <django.forms.fields.ChoiceField object>, 'name': <django.forms.fields.CharField object>, 'required': <django.forms.fields.BooleanField object>}

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField object>, 'model': <django.forms.fields.ChoiceField object>, 'name': <django.forms.fields.CharField object>, 'required': <django.forms.fields.BooleanField object>}

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''),
('initial', ''), ('required', False)]
```

48.4.1.1.1.567 fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.settings module

48.4.1.1.1.568 Module contents

48.4.1.1.1.569 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max package

48.4.1.1.1.570 Submodules

48.4.1.1.1.571 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.apps module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.apps.Config(app_name,
                                                                                       app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_select_multiple_with_max'
    name = 'fobi.contrib.plugins.form_elements.fields.select_multiple_with_max'
```

48.4.1.1.1.572 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base.SelectMultipleWithMaxInput
    Bases: fobi.base.FormFieldPlugin
    Select multiple with max field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms.
        SelectMultipleWithMaxInputForm

    get_choices()
        Get choices.

        Might be used in integration plugins.

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'

    name = 'Select multiple with max'

    prepare_plugin_form_data(cleaned_data)
        Prepare plugin form data.

        Might be used in integration plugins.

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

        Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
```

- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'select_multiple_with_max'

48.4.1.1.1.573 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.conf module

`fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from *fobi.contrib.plugins.form_elements.fields.select_multiple_with_max* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.574 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.defaults module

48.4.1.1.1.575 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fields module

class `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fields.MultipleChoiceWithMaxFi`

Bases: `django.forms.fields.MultipleChoiceField`

Multiple choice with max field.

validate(*value*)

Validate.

48.4.1.1.1.576 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements.SelectMulti`

Bases: `fobi.base.FormFieldPlugin`

Select multiple with max field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms.SelectMultipleWithMaxInputForm`

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

group = 'Fields'

name = 'Select multiple with max'

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_multiple_with_max'

48.4.1.1.1.577 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms.SelectMultipleWithMaxInputPluginForm
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for `SelectMultipleWithMaxInputPlugin`.

```
base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'max_choices':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

```
clean_initial()
```

Validating the initial value.

```
declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField
object>, 'label': <django.forms.fields.CharField object>, 'max_choices':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

```
property media
```

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text',
''), ('initial', ''), ('required', False), ('max_choices', '')]
```

48.4.1.1.1.578 `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.settings` module

48.4.1.1.1.579 Module contents

48.4.1.1.1.580 `fobi.contrib.plugins.form_elements.fields.slider` package

48.4.1.1.1.581 Submodules

48.4.1.1.1.582 `fobi.contrib.plugins.form_elements.fields.slider.apps` module

```
class fobi.contrib.plugins.form_elements.fields.slider.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_slider'
    name = 'fobi.contrib.plugins.form_elements.fields.slider'
```

48.4.1.1.1.583 `fobi.contrib.plugins.form_elements.fields.slider.base` module

```
class fobi.contrib.plugins.form_elements.fields.slider.base.SliderInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Slider field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.slider.forms.SliderInputForm
    get_choices()
        Get choices.
        Might be used in integration plugins.
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    get_initial()
        Get initial value.
        Might be used in integration plugins.
    group = 'Fields'
    html_classes = ['slider']
    name = 'Slider'
    uid = 'slider'
```


48.4.1.1.1.584 fobi.contrib.plugins.form_elements.fields.slider.conf module

`fobi.contrib.plugins.form_elements.fields.slider.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from *fobi.contrib.plugins.form_elements.fields.slider* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name.
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.585 fobi.contrib.plugins.form_elements.fields.slider.constants module

48.4.1.1.1.586 fobi.contrib.plugins.form_elements.fields.slider.defaults module

48.4.1.1.1.587 fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements.SliderInputPlugin(user=None)`

Bases: *fobi.base.FormFieldPlugin*

Slider field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.slider.forms.SliderInputForm*

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)

Get form field instances.

get_initial()

Get initial value.

Might be used in integration plugins.

group = 'Fields'

html_classes = ['slider']

name = 'Slider'

uid = 'slider'

48.4.1.1.1.588 fobi.contrib.plugins.form_elements.fields.slider.forms module

```
class fobi.contrib.plugins.form_elements.fields.slider.forms.SliderInputForm(data=None,
                                                                              files=None,
                                                                              auto_id='id_%s',
                                                                              prefix=None,
                                                                              initial=None,
                                                                              error_class=django.forms.utils.ErrorList,
                                                                              label_suffix=None,
                                                                              empty_permitted=False,
                                                                              field_order=None,
                                                                              use_required_attribute=None,
                                                                              renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for SliderInputPlugin.

```
base_fields = {'custom_ticks': <django.forms.fields.CharField object>, 'handle':
<django.forms.fields.ChoiceField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial':
<django.forms.fields.IntegerField object>, 'label': <django.forms.fields.CharField
object>, 'label_end': <django.forms.fields.CharField object>, 'label_start':
<django.forms.fields.CharField object>, 'max_value':
<django.forms.fields.IntegerField object>, 'min_value':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>,
'show_endpoints_as': <django.forms.fields.ChoiceField object>, 'step':
<django.forms.fields.IntegerField object>, 'tooltip':
<django.forms.fields.ChoiceField object>}
```

`clean()`

Validating the values.

```
declared_fields = {'custom_ticks': <django.forms.fields.CharField object>,
'handle': <django.forms.fields.ChoiceField object>, 'help_text':
<django.forms.fields.CharField object>, 'initial':
<django.forms.fields.IntegerField object>, 'label': <django.forms.fields.CharField
object>, 'label_end': <django.forms.fields.CharField object>, 'label_start':
<django.forms.fields.CharField object>, 'max_value':
<django.forms.fields.IntegerField object>, 'min_value':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>,
'show_endpoints_as': <django.forms.fields.ChoiceField object>, 'step':
<django.forms.fields.IntegerField object>, 'tooltip':
<django.forms.fields.ChoiceField object>}
```

`property media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('initial', 50), ('min_value',
0), ('max_value', 100), ('step', 1), ('tooltip', 'show'), ('handle', 'round'),
('show_endpoints_as', 'labels'), ('label_start', ''), ('label_end', ''),
('custom_ticks', ''), ('help_text', ''), ('required', False)]
```

48.4.1.1.1.589 fobi.contrib.plugins.form_elements.fields.slider.helpers module

fobi.contrib.plugins.form_elements.fields.slider.helpers.**generate_ticks**(*choices*,
empty_labels=False)

Generate ticks.

Parameters

- **choices** (*iterable*) – Iterable of tuples or lists:
- **empty_labels** (*bool*) –

Return dict

48.4.1.1.1.590 fobi.contrib.plugins.form_elements.fields.slider.settings module

48.4.1.1.1.591 fobi.contrib.plugins.form_elements.fields.slider.widgets module

class fobi.contrib.plugins.form_elements.fields.slider.widgets.**BaseSliderPluginWidget**(*plugin*)
Bases: *fobi.base.FormElementPluginWidget*

Base date form element plugin widget.

html_classes = ['slider']

plugin_uid = 'slider'

48.4.1.1.1.592 Module contents

48.4.1.1.1.593 fobi.contrib.plugins.form_elements.fields.slug package

48.4.1.1.1.594 Submodules

48.4.1.1.1.595 fobi.contrib.plugins.form_elements.fields.slug.apps module

class fobi.contrib.plugins.form_elements.fields.slug.apps.**Config**(*app_name*, *app_module*)
Bases: *django.apps.config.AppConfig*

Config.

label = 'fobi_contrib_plugins_form_elements_fields_slug'

name = 'fobi.contrib.plugins.form_elements.fields.slug'

48.4.1.1.1.596 fobi.contrib.plugins.form_elements.fields.slug.base module

```
class fobi.contrib.plugins.form_elements.fields.slug.base.SlugInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Slug field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Slug'
    uid = 'slug'
```

48.4.1.1.1.597 fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Slug field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Slug'
    uid = 'slug'
```

48.4.1.1.1.598 fobi.contrib.plugins.form_elements.fields.slug.forms module

```
class fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           prefix=None,
                                                                           initial=None,
                                                                           error_class=<class
                                                                           'django.forms.utils.ErrorList'>,
                                                                           label_suffix=None,
                                                                           empty_permitted=False,
                                                                           field_order=None,
                                                                           use_required_attribute=None,
                                                                           renderer=None)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm

    Form for SlugInputPlugin.
```

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.SlugField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

clean()

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.SlugField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('max_length', '255'), ('required', False), ('placeholder', '')]
```

48.4.1.1.1.599 Module contents

48.4.1.1.1.600 `fobi.contrib.plugins.form_elements.fields.text` package

48.4.1.1.1.601 Submodules

48.4.1.1.1.602 `fobi.contrib.plugins.form_elements.fields.text.apps` module

```
class fobi.contrib.plugins.form_elements.fields.text.apps.Config(app_name, app_module)
```

Bases: `django.apps.config AppConfig`

`Config`.

```
label = 'fobi_contrib_plugins_form_elements_fields_text'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.text'
```

48.4.1.1.1.603 `fobi.contrib.plugins.form_elements.fields.text.base` module

```
class fobi.contrib.plugins.form_elements.fields.text.base.TextInputPlugin(user=None)
```

Bases: `fobi.base.ModelFormPlugin`

Text field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm`

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
```

Get form field instances.

```
group = 'Fields'
```

```
name = 'Text'
uid = 'text'
```

48.4.1.1.1.604 fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Text field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Text'
    uid = 'text'
```

48.4.1.1.1.605 fobi.contrib.plugins.form_elements.fields.text.forms module

```
class fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           prefix=None,
                                                                           initial=None,
                                                                           error_class=<class
                                                                           'django.forms.utils.ErrorList'>,
                                                                           label_suffix=None,
                                                                           empty_permitted=False,
                                                                           field_order=None,
                                                                           use_required_attribute=None,
                                                                           renderer=None)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
    Form for TextInputPlugin.

    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
    <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
    object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
    <django.forms.fields.CharField object>, 'placeholder':
    <django.forms.fields.CharField object>, 'required':
    <django.forms.fields.BooleanField object>}

    clean()
        Validation.

    declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
    <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
    object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
    <django.forms.fields.CharField object>, 'placeholder':
    <django.forms.fields.CharField object>, 'required':
    <django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('max_length', '255'), ('required', False), ('placeholder', '')]
```

48.4.1.1.1.606 Module contents

48.4.1.1.1.607 fobi.contrib.plugins.form_elements.fields.textarea package

48.4.1.1.1.608 Submodules

48.4.1.1.1.609 fobi.contrib.plugins.form_elements.fields.textarea.apps module

```
class fobi.contrib.plugins.form_elements.fields.textarea.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_textarea'
    name = 'fobi.contrib.plugins.form_elements.fields.textarea'
```

48.4.1.1.1.610 fobi.contrib.plugins.form_elements.fields.textarea.base module

```
class fobi.contrib.plugins.form_elements.fields.textarea.base.TextareaPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Textarea field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'Textarea'
    uid = 'textarea'
```

48.4.1.1.1.611 fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements.TextareaPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Textarea field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
```

```
name = 'Textarea'
uid = 'textarea'
```

48.4.1.1.1.612 fobi.contrib.plugins.form_elements.fields.textarea.forms module

```
class fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm(data=None,
                                                                              files=None,
                                                                              auto_id='id_%s',
                                                                              prefix=None,
                                                                              initial=None, error_class=<class
                                                                              'django.forms.utils.ErrorList'>,
                                                                              label=
                                                                              bel_suffix=None,
                                                                              empty_permitted=False,
                                                                              field_order=None,
                                                                              use_required_attribute=None,
                                                                              renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for TextareaPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':
<django.forms.fields.CharField object>, 'placeholder':
<django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('required', False), ('max_length', ''), ('placeholder', '')]
```

48.4.1.1.1.613 Module contents

48.4.1.1.1.614 fobi.contrib.plugins.form_elements.fields.time package

48.4.1.1.1.615 Submodules

48.4.1.1.1.616 fobi.contrib.plugins.form_elements.fields.time.apps module

```
class fobi.contrib.plugins.form_elements.fields.time.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
```


Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_time'
name = 'fobi.contrib.plugins.form_elements.fields.time'
```

48.4.1.1.1.617 fobi.contrib.plugins.form_elements.fields.time.base module

```
class fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Time field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Time'

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'time'
```

48.4.1.1.1.618 fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Time field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Fields'
    name = 'Time'

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –
```

```
uid = 'time'
```

48.4.1.1.1.619 fobi.contrib.plugins.form_elements.fields.time.forms module

```
class fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           prefix=None,
                                                                           initial=None,
                                                                           error_class=<class
                                                                           'django.forms.utils.ErrorList'>,
                                                                           label_suffix=None,
                                                                           empty_permitted=False,
                                                                           field_order=None,
                                                                           use_required_attribute=None,
                                                                           renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for TimeInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.TimeField object>, 'input_formats':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
clean_initial()
```

Clean the initial value.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':
<django.forms.fields.TimeField object>, 'input_formats':
<django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField
object>, 'name': <django.forms.fields.CharField object>, 'required':
<django.forms.fields.BooleanField object>}
```

```
property media
```

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',
''), ('input_formats', ''), ('required', False)]
```

48.4.1.1.1.620 Module contents

48.4.1.1.1.621 fobi.contrib.plugins.form_elements.fields.url package

48.4.1.1.1.622 Submodules

48.4.1.1.1.623 fobi.contrib.plugins.form_elements.fields.url.apps module

```
class fobi.contrib.plugins.form_elements.fields.url.apps.Config(app_name, app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_url'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.url'
```

48.4.1.1.1.624 fobi.contrib.plugins.form_elements.fields.url.base module

```
class fobi.contrib.plugins.form_elements.fields.url.base.URLInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    URL input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'URL'
    uid = 'url'
```

48.4.1.1.1.625 fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    URL input plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = 'Fields'
    name = 'URL'
    uid = 'url'
```

48.4.1.1.1.626 fobi.contrib.plugins.form_elements.fields.url.forms module

```
class fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm(data=None, files=None,
    auto_id='id_%s',
    prefix=None,
    initial=None,
    error_class=<class
        'django.forms.utils.ErrorList'>,
    label_suffix=None,
    empty_permitted=False,
    field_order=None,
    use_required_attribute=None,
    renderer=None)
    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
    Form for URLPlugin.
```

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':  
<django.forms.fields.URLField object>, 'label': <django.forms.fields.CharField  
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':  
<django.forms.fields.CharField object>, 'placeholder':  
<django.forms.fields.CharField object>, 'required':  
<django.forms.fields.BooleanField object>}
```

clean()

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial':  
<django.forms.fields.URLField object>, 'label': <django.forms.fields.CharField  
object>, 'max_length': <django.forms.fields.IntegerField object>, 'name':  
<django.forms.fields.CharField object>, 'placeholder':  
<django.forms.fields.CharField object>, 'required':  
<django.forms.fields.BooleanField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial',  
''), ('max_length', '255'), ('required', False), ('placeholder', '')]
```

48.4.1.1.1.627 Module contents

48.4.1.1.1.628 Module contents

48.4.1.1.1.629 `fobi.contrib.plugins.form_elements.security` package

48.4.1.1.1.630 Subpackages

48.4.1.1.1.631 `fobi.contrib.plugins.form_elements.security.captcha` package

48.4.1.1.1.632 Submodules

48.4.1.1.1.633 `fobi.contrib.plugins.form_elements.security.captcha.apps` module

```
class fobi.contrib.plugins.form_elements.security.captcha.apps.Config(app_name, app_module)  
    Bases: django.apps.config.AppConfig  
    Config.  
    label = 'fobi_contrib_plugins_form_elements_security_captcha'  
    name = 'fobi.contrib.plugins.form_elements.security.captcha'
```

48.4.1.1.1.634 fobi.contrib.plugins.form_elements.security.captcha.base module

```
class fobi.contrib.plugins.form_elements.security.captcha.base.CaptchaInputPlugin(user=None)
    Bases: fobi.base.FormElementPlugin
    Captcha field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Security'
    name = 'Captcha'
    uid = 'captcha'
```

48.4.1.1.1.635 fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaInputPlugin(user=None)
    Bases: fobi.base.FormElementPlugin
    Captcha field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Security'
    name = 'Captcha'
    uid = 'captcha'
```

48.4.1.1.1.636 fobi.contrib.plugins.form_elements.security.captcha.forms module

```
class fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaInputForm(data=None,
    files=None,
    auto_id='id_%s',
    pre-
    fix=None,
    ini-
    tial=None,
    er-
    ror_class=<class
    'django.forms.utils.ErrorList'
    la-
    bel_suffix=None,
    empty_permitted=False,
    field_order=None,
    use_required_attribute=None,
    ren-
    derer=None)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
```

Form for CaptchaInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}

property media
    Return all media required to render the widgets on this form.

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('required',
True)]
```

48.4.1.1.1.637 Module contents

48.4.1.1.1.638 fobi.contrib.plugins.form_elements.security.honeypot package

48.4.1.1.1.639 Submodules

48.4.1.1.1.640 fobi.contrib.plugins.form_elements.security.honeypot.apps module

```
class fobi.contrib.plugins.form_elements.security.honeypot.apps.Config(app_name,
                                                                    app_module)

    Bases: django.apps.config.AppConfig

    Config.

    label = 'fobi_contrib_plugins_form_elements_security_honeypot'
    name = 'fobi.contrib.plugins.form_elements.security.honeypot'
```

48.4.1.1.1.641 fobi.contrib.plugins.form_elements.security.honeypot.base module

```
class fobi.contrib.plugins.form_elements.security.honeypot.base.HoneypotInputPlugin(user=None)
    Bases: fobi.base.FormElementPlugin

    Honeypot field plugin.

    form
        alias          of          fobi.contrib.plugins.form_elements.security.honeypot.forms.
        HoneypotInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Security'

    is_hidden = True

    name = 'Honeypot'

    uid = 'honeypot'
```

48.4.1.1.1.642 fobi.contrib.plugins.form_elements.security.honeypot.conf module

`fobi.contrib.plugins.form_elements.security.honeypot.conf.get_setting(setting, override=None)`
Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.security.honeypot.conf` module, falling back to the default.

If `override` is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

48.4.1.1.1.643 fobi.contrib.plugins.form_elements.security.honeypot.defaults module

48.4.1.1.1.644 fobi.contrib.plugins.form_elements.security.honeypot.fields module

```
class fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotField(*,
                                                                              max_length=None,
                                                                              min_length=None,
                                                                              strip=True,
                                                                              empty_value="",
                                                                              **kwargs)
```

Bases: `django.forms.fields.CharField`

clean(*value*)

Check that honeypot value remained the same.

default_error_messages = {'invalid': 'Field value was tampered with.'}

widget

alias of `django.forms.widgets.HiddenInput`

48.4.1.1.1.645 fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin(user=None)
```

Bases: `fobi.base.FormElementPlugin`

Honeypot field plugin.

form

alias of `fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotInputForm`

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
Get form field instances.

group = 'Security'

is_hidden = True

name = 'Honeypot'

uid = 'honeypot'

48.4.1.1.1.646 fobi.contrib.plugins.form_elements.security.honeypot.forms module

```
class fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotInputForm(data=None,
                                          files=None,
                                          auto_id='id_%s',
                                          pre-
                                          fix=None,
                                          ini-
                                          tial=None,
                                          er-
                                          ror_class=<class
                                          'django.forms.utils.ErrorL
                                          la-
                                          bel_suffix=None,
                                          empty_permitted=False,
                                          field_order=None,
                                          use_required_attribute=No
                                          ren-
                                          derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for HoneypotInputPlugin.

```
base_fields = {'initial': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'max_length':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

```
declared_fields = {'initial': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'max_length':
<django.forms.fields.IntegerField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

property `media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('initial', ''), ('max_length',
'255'), ('required', True)]
```

48.4.1.1.1.647 fobi.contrib.plugins.form_elements.security.honeypot.settings module

- `HONEYPOT_VALUE` (string)

48.4.1.1.1.648 Module contents

48.4.1.1.1.649 fobi.contrib.plugins.form_elements.security.recaptcha package

48.4.1.1.1.650 Submodules

48.4.1.1.1.651 fobi.contrib.plugins.form_elements.security.recaptcha.apps module

```
class fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_security_recaptcha'
    name = 'fobi.contrib.plugins.form_elements.security.recaptcha'
```

48.4.1.1.1.652 fobi.contrib.plugins.form_elements.security.recaptcha.base module

```
class fobi.contrib.plugins.form_elements.security.recaptcha.base.ReCaptchaInputPlugin(user=None)
    Bases: fobi.base.FormElementPlugin
    ReCaptcha field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.security.recaptcha.forms.
                             ReCaptchaInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Security'
    name = 'ReCaptcha'
    uid = 'recaptcha'
```

48.4.1.1.1.653 fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements.ReCaptchaInputPlugin(user=None)
    Bases: fobi.base.FormElementPlugin
    ReCaptcha field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.security.recaptcha.forms.
                             ReCaptchaInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Security'
    name = 'ReCaptcha'
    uid = 'recaptcha'
```

48.4.1.1.1.654 fobi.contrib.plugins.form_elements.security.recaptcha.forms module

```
class fobi.contrib.plugins.form_elements.security.recaptcha.forms.ReCaptchaInputForm(data=None,
                                          files=None,
                                          auto_id='id_%s',
                                          pre-
                                          fix=None,
                                          ini-
                                          tial=None,
                                          er-
                                          ror_class=<class
                                          'django.forms.utils.Error
                                          la-
                                          bel_suffix=None,
                                          empty_permitted=False,
                                          field_order=None,
                                          use_required_attribute=
                                          ren-
                                          derer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for ReCaptchaInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'label':
<django.forms.fields.CharField object>, 'name': <django.forms.fields.CharField
object>, 'required': <django.forms.fields.BooleanField object>}
```

property `media`

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('required',
True)]
```

48.4.1.1.1.655 Module contents

48.4.1.1.1.656 Module contents

48.4.1.1.1.657 fobi.contrib.plugins.form_elements.test package

48.4.1.1.1.658 Subpackages

48.4.1.1.1.659 fobi.contrib.plugins.form_elements.test.dummy package

48.4.1.1.1.660 Submodules

48.4.1.1.1.661 fobi.contrib.plugins.form_elements.test.dummy.apps module

```
class fobi.contrib.plugins.form_elements.test.dummy.apps.Config(app_name, app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_test_dummy'
name = 'fobi.contrib.plugins.form_elements.test.dummy'
```

48.4.1.1.1.662 fobi.contrib.plugins.form_elements.test.dummy.base module

```
class fobi.contrib.plugins.form_elements.test.dummy.base.DummyPlugin(user=None)
    Bases: fobi.base.FormElementPlugin

    Dummy plugin.

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Testing'
    name = 'Dummy'
    post_processor()
        Post process data.

        Always the same.

    uid = 'dummy'
```

48.4.1.1.1.663 fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin(user=None)
    Bases: fobi.base.FormElementPlugin

    Dummy plugin.

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    group = 'Testing'
    name = 'Dummy'
    post_processor()
        Post process data.

        Always the same.

    uid = 'dummy'
```

48.4.1.1.1.664 fobi.contrib.plugins.form_elements.test.dummy.widgets module

```
class fobi.contrib.plugins.form_elements.test.dummy.widgets.BaseDummyPluginWidget(plugin)
    Bases: fobi.base.FormElementPluginWidget

    Base dummy form element plugin widget.

    plugin_uid = 'dummy'
```

48.4.1.1.1.665 Module contents

48.4.1.1.1.666 Module contents

48.4.1.1.1.667 Module contents

48.4.1.1.1.668 fobi.contrib.plugins.form_handlers package

48.4.1.1.1.669 Subpackages

48.4.1.1.1.670 fobi.contrib.plugins.form_handlers.db_store package

48.4.1.1.1.671 Subpackages

48.4.1.1.1.672 fobi.contrib.plugins.form_handlers.db_store.migrations package

48.4.1.1.1.673 Submodules

48.4.1.1.1.674 fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial module

```
class fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial.Migration(name,  
                                                                    app_label)
```

Bases: django.db.migrations.migration.Migration

```
dependencies = [('fobi', '0002_auto_20150912_1744'), ('auth', '__first__')]
```

```
operations = [<CreateModel name='SavedFormDataEntry', fields=[('id',  
<django.db.models.fields.AutoField>), ('form_data_headers',  
<django.db.models.fields.TextField>), ('saved_data',  
<django.db.models.fields.TextField>), ('created',  
<django.db.models.fields.DateTimeField>), ('form_entry',  
<django.db.models.fields.related.ForeignKey>), ('user',  
<django.db.models.fields.related.ForeignKey>)], options={'verbose_name': 'Saved  
form data entry', 'abstract': False, 'verbose_name_plural': 'Saved form data  
entries', 'db_table': 'db_store_savedformdataentry'}>]
```

48.4.1.1.1.675 fobi.contrib.plugins.form_handlers.db_store.migrations.0002_savedformwizardddataentry module

```
class fobi.contrib.plugins.form_handlers.db_store.migrations.0002_savedformwizardddataentry.Migration(name,  
                                                                    app_label)
```

Bases: django.db.migrations.migration.Migration

```
dependencies = [('auth', '__first__'), ('fobi', '0010_formwizardhandler'),  
               ('fobi_contrib_plugins_form_handlers_db_store', '0001_initial')]
```

```
operations = [<CreateModel name='SavedFormWizardDataEntry', fields=[('id',
<django.db.models.fields.AutoField>), ('form_data_headers',
<django.db.models.fields.TextField>), ('saved_data',
<django.db.models.fields.TextField>), ('created',
<django.db.models.fields.DateTimeField>), ('form_wizard_entry',
<django.db.models.fields.related.ForeignKey>), ('user',
<django.db.models.fields.related.ForeignKey>)], options={'abstract': False,
'db_table': 'db_store_savedformwizarddataentry', 'verbose_name': 'Saved form
wizard data entry', 'verbose_name_plural': 'Saved form wizard data entries'}>]
```

48.4.1.1.1.676 Module contents

48.4.1.1.1.677 fobi.contrib.plugins.form_handlers.db_store.urls package

48.4.1.1.1.678 Submodules

48.4.1.1.1.679 fobi.contrib.plugins.form_handlers.db_store.urls.form_handlers module

48.4.1.1.1.680 fobi.contrib.plugins.form_handlers.db_store.urls.form_wizard_handlers module

48.4.1.1.1.681 Module contents

48.4.1.1.1.682 Submodules

48.4.1.1.1.683 fobi.contrib.plugins.form_handlers.db_store.admin module

```
class fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDataEntryAdmin(model,
                                                                                   admin_site)
    Bases: fobi.contrib.plugins.form_handlers.db_store.admin.BaseSavedFormDataEntryAdmin
    Saved form data entry admin.

    class Meta
        Bases: object
        Meta class.

        app_label = 'Saved form data entry'

    actions = ['export_data']

    fieldsets = ((None, {'fields': ('form_entry', 'user')}), ('Data', {'fields':
('formatted_saved_data', 'created')}), ('Raw', {'classes': ('collapse',), 'fields':
('form_data_headers', 'saved_data')}))

    list_display = ('form_entry', 'user', 'formatted_saved_data', 'created')

    list_filter = ('form_entry', 'user')

    property media

    only_args = ['form_entry']

    readonly_fields = ('created', 'formatted_saved_data')
```

```
class fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntryAdmin(model,
                                                                                      ad-
                                                                                      min_site)

Bases: fobi.contrib.plugins.form_handlers.db_store.admin.BaseSavedFormDataEntryAdmin
Saved form wizard data entry admin.

class Meta
    Bases: object
    Meta class.
    app_label = 'Saved form wizard data entry'
actions = ['export_data']
fieldsets = ((None, {'fields': ('form_wizard_entry', 'user')}), ('Data', {'fields':
('formatted_saved_data', 'created')}), ('Raw', {'classes': ('collapse',), 'fields':
('form_data_headers', 'saved_data')}))
list_display = ('form_wizard_entry', 'user', 'formatted_saved_data', 'created')
list_filter = ('form_wizard_entry', 'user')
property media
only_args = ['form_wizard_entry']
readonly_fields = ('created', 'formatted_saved_data')
```

48.4.1.1.1.684 fobi.contrib.plugins.form_handlers.db_store.apps module

```
class fobi.contrib.plugins.form_handlers.db_store.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_handlers_db_store'
    name = 'fobi.contrib.plugins.form_handlers.db_store'
```

48.4.1.1.1.685 fobi.contrib.plugins.form_handlers.db_store.base module

```
class fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin(user=None)
    Bases: fobi.base.FormHandlerPlugin
    DB store form handler plugin.
    Can be used only once per form.
    allow_multiple = False
    custom_actions(form_entry, request=None)
        Custom actions.
        Adding a link to view the saved form entries.
        Return iterable
    name = 'DB store'
    run(form_entry, request, form, form_element_entries=None)
        Run.
```

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

save_form_data_entry(*form_entry, request, field_name_to_label_map, cleaned_data*)

Save form data entry.

Might be used in integration plugins.

uid = 'db_store'

class `fobi.contrib.plugins.form_handlers.db_store.base.DBStoreWizardHandlerPlugin`(*user=None*)

Bases: `fobi.base.FormWizardHandlerPlugin`

DB store form wizard handler plugin.

Can be used only once per form.

allow_multiple = **False**

custom_actions(*form_wizard_entry, request=None*)

Custom actions.

Adding a link to view the saved form entries.

Return iterable

name = 'DB store'

run(*form_wizard_entry, request, form_list, form_wizard, form_element_entries=None*)

Run.

Parameters

- **form_wizard_entry** (`fobi.models.FormWizardEntry`) – Instance of `fobi.models.FormWizardEntry`.
- **request** (`django.http.HttpRequest`) –
- **form_list** (*list*) – List of `django.forms.Form` instances.
- **form_wizard** (`fobi.wizard.views.dynamic.DynamicWizardView`) – Instance of `fobi.wizard.views.dynamic.DynamicWizardView`.
- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

uid = 'db_store'

48.4.1.1.1.686 fobi.contrib.plugins.form_handlers.db_store.callbacks module

class fobi.contrib.plugins.form_handlers.db_store.callbacks.AutoFormDbStore

Bases: *fobi.base.FormCallback*

Auto save form entries.

Note, that this callback is not active. In order to activate it, you should import the AutoFormDbStore and register it using the callback register as follows.

```
>>> from fobi.base import form_callback_registry
>>> from fobi.contrib.plugins.form_handlers.db_store.callbacks import (
>>>     AutoFormDbStore
>>> )
>>> form_callback_registry.register(AutoFormDbStore)
```

callback(*form_entry, request, form*)

Callback.

Parameters

- **form_entry** –
- **request** –
- **form** –

Returns

stage = 'form_valid'

48.4.1.1.1.687 fobi.contrib.plugins.form_handlers.db_store.conf module

fobi.contrib.plugins.form_handlers.db_store.conf.get_setting(*setting, override=None*)

Get setting.

Get a setting from fobi.contrib.plugins.form_handlers.db_store conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.1.1.1.688 fobi.contrib.plugins.form_handlers.db_store.defaults module

48.4.1.1.1.689 fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers module

class fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreHandlerPlugin(*user=None*)

Bases: *fobi.base.FormHandlerPlugin*

DB store form handler plugin.

Can be used only once per form.

allow_multiple = False

custom_actions(*form_entry*, *request=None*)

Custom actions.

Adding a link to view the saved form entries.

Return iterable

name = 'DB store'

run(*form_entry*, *request*, *form*, *form_element_entries=None*)

Run.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

save_form_data_entry(*form_entry*, *request*, *field_name_to_label_map*, *cleaned_data*)

Save form data entry.

Might be used in integration plugins.

uid = 'db_store'

class *fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreWizardHandlerPlugin*(*user=None*)

Bases: *fobi.base.FormWizardHandlerPlugin*

DB store form wizard handler plugin.

Can be used only once per form.

allow_multiple = False

custom_actions(*form_wizard_entry*, *request=None*)

Custom actions.

Adding a link to view the saved form entries.

Return iterable

name = 'DB store'

run(*form_wizard_entry*, *request*, *form_list*, *form_wizard*, *form_element_entries=None*)

Run.

Parameters

- **form_wizard_entry** (*fobi.models.FormWizardEntry*) – Instance of *fobi.models.FormWizardEntry*.
- **request** (*django.http.HttpRequest*) –
- **form_list** (*list*) – List of *django.forms.Form* instances.
- **form_wizard** (*fobi.wizard.views.dynamic.DynamicWizardView*) – Instance of *fobi.wizard.views.dynamic.DynamicWizardView*.
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

```
uid = 'db_store'
```

48.4.1.1.1.690 fobi.contrib.plugins.form_handlers.db_store.helpers module

```
class fobi.contrib.plugins.form_handlers.db_store.helpers.DataExporter(queryset, only_args)
```

Bases: object

Exporting the data.

```
export_to_csv()
```

Export data to CSV.

```
export_to_xls()
```

Export data to XLS.

```
graceful_export()
```

Export data into XLS/CSV depending on what is available.

48.4.1.1.1.691 fobi.contrib.plugins.form_handlers.db_store.models module

```
class fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataEntry(*args,
                                                                                     **kwargs)
```

Bases: django.db.models.base.Model

Abstract saved form data entry.

```
class Meta
```

Bases: object

Meta options.

```
abstract = False
```

```
created
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
form_data_headers
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
formatted_saved_data()
```

Shows the formatted saved data records.

Return string

```
get_next_by_created(*, field=<django.db.models.fields.DateTimeField: created>, is_next=True,
                    **kwargs)
```

```
get_previous_by_created(*, field=<django.db.models.fields.DateTimeField: created>, is_next=False,
                        **kwargs)
```

```
saved_data
```

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
user
```

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOne-ToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
class fobi.contrib.plugins.form_handlers.db_store.models.SavedFormDataEntry(*args,
                                                                            **kwargs)
```

Bases: *fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataEntry*

Saved form data.

exception DoesNotExist

Bases: *django.core.exceptions.ObjectDoesNotExist*

exception MultipleObjectsReturned

Bases: *django.core.exceptions.MultipleObjectsReturned*

form_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

form_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

```
get_next_by_created(*, field=<django.db.models.fields.DateTimeField: created>, is_next=True,
                    **kwargs)
```

```
get_previous_by_created(*, field=<django.db.models.fields.DateTimeField: created>, is_next=False,
                        **kwargs)
```

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

class fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardDataEntry(*args, **kwargs)

Bases: *fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataEntry*

Saved form data.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

form_wizard_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

form_wizard_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

get_next_by_created(*args, field=<django.db.models.fields.DateTimeField: created>, is_next=True, **kwargs)

get_previous_by_created(*args, field=<django.db.models.fields.DateTimeField: created>, is_next=False, **kwargs)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

48.4.1.1.1.692 fobi.contrib.plugins.form_handlers.db_store.settings module

- `CSV_DELIMITER` (string)
- `CSV_QUOTECHAR` (string)

48.4.1.1.1.693 fobi.contrib.plugins.form_handlers.db_store.views module

`fobi.contrib.plugins.form_handlers.db_store.views.export_saved_form_data_entries(request, form_entry_id=None, theme=None)`

Export saved form data entries.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_entry_id** (*int*) – Form ID.
- **theme** (*fobi.base.BaseTheme*) – Subclass of `fobi.base.BaseTheme`.

Return `django.http.HttpResponse`

`fobi.contrib.plugins.form_handlers.db_store.views.export_saved_form_wizard_data_entries(request, form_wizard_entry_id=None, theme=None)`

Export saved form wizard data entries.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_wizard_entry_id** (*int*) – Form ID.
- **theme** (*fobi.base.BaseTheme*) – Subclass of `fobi.base.BaseTheme`.

Return `django.http.HttpResponse`

`fobi.contrib.plugins.form_handlers.db_store.views.view_saved_form_data_entries(request, form_entry_id=None, theme=None, template_name='db_store/view_saved_form_data_entries.html')`

View saved form data entries.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_entry_id** (*int*) – Form ID.
- **theme** (*fobi.base.BaseTheme*) – Subclass of `fobi.base.BaseTheme`.
- **template_name** (*string*) –

Return `django.http.HttpResponse`

`fobi.contrib.plugins.form_handlers.db_store.views.view_saved_form_wizard_data_entries(request, form_wizard_entry_id=None, theme=None, template_name='db_store/view_saved_form_wizard_data_entries.html')`

View saved form wizard data entries.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_wizard_entry_id** (*int*) – Form ID.
- **theme** (*fobi.base.BaseTheme*) – Subclass of *fobi.base.BaseTheme*.
- **template_name** (*string*) –

Return *django.http.HttpResponse*

48.4.1.1.1.694 fobi.contrib.plugins.form_handlers.db_store.widgets module

class *fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget(plugin)*

Bases: *fobi.base.FormHandlerPluginWidget*

Base dummy form element plugin widget.

export_entries_icon_class = 'glyphicon glyphicon-export'

plugin_uid = 'db_store'

view_entries_icon_class = 'glyphicon glyphicon-list'

view_saved_form_data_entries_template_name =
'db_store/view_saved_form_data_entries.html'

48.4.1.1.1.695 Module contents

48.4.1.1.1.696 fobi.contrib.plugins.form_handlers.http_repost package

48.4.1.1.1.697 Submodules

48.4.1.1.1.698 fobi.contrib.plugins.form_handlers.http_repost.apps module

class *fobi.contrib.plugins.form_handlers.http_repost.apps.Config(app_name, app_module)*

Bases: *django.apps.config AppConfig*

Config.

label = 'fobi_contrib_plugins_form_handlers_http_repost'

name = 'fobi.contrib.plugins.form_handlers.http_repost'

48.4.1.1.1.699 fobi.contrib.plugins.form_handlers.http_repost.base module

class *fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandlerPlugin(user=None)*

Bases: *fobi.base.FormHandlerPlugin*

HTTP repost handler plugin.

Makes a HTTP repost to a given endpoint. Should be executed before *db_store* plugin.

do_http_repost(*request, files*)

Re-post data via HTTP.

Might be used in integration plugins.

form
alias of *fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm*

name = 'HTTP Repost'

plugin_data_repr()

Human readable representation of plugin data.

Return string

run(*form_entry, request, form, form_element_entries=None*)

Run.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

uid = 'http_repost'

class *fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostWizardHandlerPlugin*(*user=None*)

Bases: *fobi.base.FormWizardHandlerPlugin*

HTTP repost wizard handler plugin.

Makes a HTTP repost to a given endpoint. Should be executed before *db_store* plugin.

form

alias of *fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm*

name = 'HTTP Repost'

plugin_data_repr()

Human readable representation of plugin data.

Return string

run(*form_wizard_entry, request, form_list, form_wizard, form_element_entries=None*)

Run.

Parameters

- **form_wizard_entry** (*fobi.models.FormWizardEntry*) – Instance of *fobi.models.FormWizardEntry*.
- **request** (*django.http.HttpRequest*) –
- **form_list** (*list*) – List of *django.forms.Form* instances.
- **form_wizard** (*fobi.wizard.views.dynamic.DynamicWizardView*) – Instance of *fobi.wizard.views.dynamic.DynamicWizardView*.
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

uid = 'http_repost'

48.4.1.1.1.700 fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers module

class `fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostHandlerPlugin`(*user=None*)

Bases: `fobi.base.FormHandlerPlugin`

HTTP repost handler plugin.

Makes a HTTP repost to a given endpoint. Should be executed before `db_store` plugin.

do_http_repost(*request, files*)

Re-post data via HTTP.

Might be used in integration plugins.

form

alias of `fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm`

name = 'HTTP Repost'

plugin_data_repr()

Human readable representation of plugin data.

Return string

run(*form_entry, request, form, form_element_entries=None*)

Run.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

uid = 'http_repost'

class `fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostWizardHandlerPlugin`(*user=None*)

Bases: `fobi.base.FormWizardHandlerPlugin`

HTTP repost wizard handler plugin.

Makes a HTTP repost to a given endpoint. Should be executed before `db_store` plugin.

form

alias of `fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm`

name = 'HTTP Repost'

plugin_data_repr()

Human readable representation of plugin data.

Return string

run(*form_wizard_entry, request, form_list, form_wizard, form_element_entries=None*)

Run.

Parameters

- **form_wizard_entry** (`fobi.models.FormWizardEntry`) – Instance of `fobi.models.FormWizardEntry`.
- **request** (`django.http.HttpRequest`) –

- **form_list** (*list*) – List of `django.forms.Form` instances.
- **form_wizard** (`fobi.wizard.views.dynamic.DynamicWizardView`) – Instance of `fobi.wizard.views.dynamic.DynamicWizardView`.
- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

```
uid = 'http_repost'
```

48.4.1.1.1.701 fobi.contrib.plugins.form_handlers.http_repost.forms module

```
class fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           prefix=None,
                                                                           initial=None,
                                                                           error_class=<class
                                                                           'django.forms.utils.ErrorList'>,
                                                                           label_suffix=None,
                                                                           empty_permitted=False,
                                                                           field_order=None,
                                                                           use_required_attribute=None,
                                                                           renderer=None)
```

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

Form for HTTPRepostPlugin.

```
base_fields = {'endpoint_url': <django.forms.fields.URLField object>}
```

```
declared_fields = {'endpoint_url': <django.forms.fields.URLField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('endpoint_url', '')]
```

48.4.1.1.1.702 Module contents

48.4.1.1.1.703 fobi.contrib.plugins.form_handlers.mail package

48.4.1.1.1.704 Submodules

48.4.1.1.1.705 fobi.contrib.plugins.form_handlers.mail.apps module

```
class fobi.contrib.plugins.form_handlers.mail.apps.Config(app_name, app_module)
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_handlers_mail'
```

```
name = 'fobi.contrib.plugins.form_handlers.mail'
```

48.4.1.1.1.706 fobi.contrib.plugins.form_handlers.mail.base module

```
class fobi.contrib.plugins.form_handlers.mail.base.MailHandlerPlugin(user=None)
    Bases: fobi.base.FormHandlerPlugin, fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin
    Mail handler plugin.
    Sends emails to the person specified. Should be executed before db_store and http_repost plugins.

    form
        alias of fobi.contrib.plugins.form_handlers.mail.forms.MailForm
    name = 'Mail'
    plugin_data_repr()
        Human readable representation of plugin data.
        Return string
    run(form_entry, request, form, form_element_entries=None)
        Run.
        Parameters
            • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
            • request (django.http.HttpRequest) –
            • form (django.forms.Form) –
            • form_element_entries (iterable) – Iterable of fobi.models.FormElementEntry objects.

    uid = 'mail'
```

```
class fobi.contrib.plugins.form_handlers.mail.base.MailWizardHandlerPlugin(user=None)
    Bases: fobi.base.FormWizardHandlerPlugin
    Mail wizard handler plugin.
    Sends emails to the person specified. Should be executed before db_store and http_repost plugins.

    form
        alias of fobi.contrib.plugins.form_handlers.mail.forms.MailForm
    name = 'Mail'
    plugin_data_repr()
        Human readable representation of plugin data.
        Return string
    run(form_wizard_entry, request, form_list, form_wizard, form_element_entries=None)
        Run.
        Parameters
            • form_wizard_entry (fobi.models.FormWizardEntry) – Instance of fobi.models.FormWizardEntry.
            • request (django.http.HttpRequest) –
            • form_list (list) – List of django.forms.Form instances.
            • form_wizard (fobi.wizard.views.dynamic.DynamicWizardView) – Instance of fobi.wizard.views.dynamic.DynamicWizardView.
```

- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

`uid = 'mail'`

48.4.1.1.1.707 `fobi.contrib.plugins.form_handlers.mail.callbacks` module

class `fobi.contrib.plugins.form_handlers.mail.callbacks.AutoFormMail`

Bases: `fobi.base.FormCallback`, `fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin`

Auto mail form entries.

Note, that this callback is not active. In order to activate it, you should import the `AutoFormMail` and register it using the callback register as follows.

```
>>> from fobi.base import form_callback_registry
>>> from fobi.contrib.plugins.form_handlers.mail.callbacks import (
>>>     AutoFormMail
>>> )
>>> form_callback_registry.register(AutoFormMail)
```

callback(*form_entry*, *request*, *form*)
Callback.

Parameters

- **form_entry** –
- **request** –
- **form** –

Returns

send_email(*rendered_data*, *files*)
Send email.

Might be used in integration packages.

stage = `'form_valid'`

48.4.1.1.1.708 `fobi.contrib.plugins.form_handlers.mail.conf` module

`fobi.contrib.plugins.form_handlers.mail.conf.get_setting`(*setting*, *override=None*)
Get setting.

Get a setting from `fobi.contrib.plugins.form_handlers.mail.conf` module, falling back to the default.

If *override* is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

48.4.1.1.1.709 fobi.contrib.plugins.form_handlers.mail.defaults module

48.4.1.1.1.710 fobi.contrib.plugins.form_handlers.mail.fields module

```
class fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField(*, required=True,
                                                                    widget=None,
                                                                    label=None, initial=None,
                                                                    help_text="",
                                                                    error_messages=None,
                                                                    show_hidden_initial=False,
                                                                    validators=(),
                                                                    localize=False,
                                                                    disabled=False,
                                                                    label_suffix=None)
```

Bases: `django.forms.fields.Field`

MultiEmailField.

code = 'invalid'

message = 'Enter valid email addresses.'

to_python(value)

Normalize data to a list of strings.

validate(value)

Check if value consists only of valid emails.

widget

alias of `fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget`

48.4.1.1.1.711 fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers module

```
class fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailHandlerPlugin(user=None)
```

Bases: `fobi.base.FormHandlerPlugin`, `fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin`

Mail handler plugin.

Sends emails to the person specified. Should be executed before `db_store` and `http_repost` plugins.

form

alias of `fobi.contrib.plugins.form_handlers.mail.forms.MailForm`

name = 'Mail'

plugin_data_repr()

Human readable representation of plugin data.

Return string

run(form_entry, request, form, form_element_entries=None)

Run.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

uid = 'mail'

class `fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailWizardHandlerPlugin`(*user=None*)
 Bases: `fobi.base.FormWizardHandlerPlugin`

Mail wizard handler plugin.

Sends emails to the person specified. Should be executed before `db_store` and `http_repost` plugins.

form

alias of `fobi.contrib.plugins.form_handlers.mail.forms.MailForm`

name = 'Mail'

plugin_data_repr()

Human readable representation of plugin data.

Return string

run(*form_wizard_entry, request, form_list, form_wizard, form_element_entries=None*)

Run.

Parameters

- **form_wizard_entry** (`fobi.models.FormWizardEntry`) – Instance of `fobi.models.FormWizardEntry`.
- **request** (`django.http.HttpRequest`) –
- **form_list** (*list*) – List of `django.forms.Form` instances.
- **form_wizard** (`fobi.wizard.views.dynamic.DynamicWizardView`) – Instance of `fobi.wizard.views.dynamic.DynamicWizardView`.
- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

uid = 'mail'

48.4.1.1.1.712 fobi.contrib.plugins.form_handlers.mail.forms module

class `fobi.contrib.plugins.form_handlers.mail.forms.MailForm`(*data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None, renderer=None*)

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

Form for BooleanSelectPlugin.

```
base_fields = {'body': <django.forms.fields.CharField object>, 'from_email':  
<django.forms.fields.EmailField object>, 'from_name':  
<django.forms.fields.CharField object>, 'subject': <django.forms.fields.CharField  
object>, 'to_email':  
<fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField object>, 'to_name':  
<django.forms.fields.CharField object>}
```

```
declared_fields = {'body': <django.forms.fields.CharField object>, 'from_email':  
<django.forms.fields.EmailField object>, 'from_name':  
<django.forms.fields.CharField object>, 'subject': <django.forms.fields.CharField  
object>, 'to_email':  
<fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField object>, 'to_name':  
<django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

```
plugin_data_fields = [('from_name', ''), ('from_email', ''), ('to_name', ''),  
('to_email', ''), ('subject', ''), ('body', '')]
```

48.4.1.1.1.713 fobi.contrib.plugins.form_handlers.mail.helpers module

```
fobi.contrib.plugins.form_handlers.mail.helpers.send_mail(subject, message, from_email,  
recipient_list, fail_silently=False,  
auth_user=None, auth_password=None,  
connection=None, html_message=None,  
attachments=None)
```

Send email.

Easy wrapper for sending a single message to a recipient list. All members of the recipient list will see the other recipients in the ‘To’ field.

If `auth_user` is `None`, the `EMAIL_HOST_USER` setting is used. If `auth_password` is `None`, the `EMAIL_HOST_PASSWORD` setting is used.

Note: The API for this method is frozen. New code wanting to extend the functionality should use the `EmailMessage` class directly.

48.4.1.1.1.714 fobi.contrib.plugins.form_handlers.mail.mixins module

```
class fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin
```

Bases: `object`

Mail handler mixin.

```
get_base_url(request)
```

Get base URL.

Might be used in integration packages.

```
get_rendered_data(cleaned_data, field_name_to_label_map, base_url)
```

Get rendered data.

Might be used in integration packages.

```
send_email(rendered_data, files)
```

Send email.

Might be used in integration packages.

48.4.1.1.1.715 fobi.contrib.plugins.form_handlers.mail.settings module

48.4.1.1.1.716 fobi.contrib.plugins.form_handlers.mail.widgets module

class fobi.contrib.plugins.form_handlers.mail.widgets.**MultiEmailWidget**(*attrs=None*)

Bases: django.forms.widgets.Textarea

Multi email widget.

is_hidden = False

property media

prep_value(*value*)

Prepare value before effectively render widget

render(*name, value, attrs=None, **kwargs*)

Render.

48.4.1.1.1.717 Module contents

48.4.1.1.1.718 Module contents

48.4.1.1.1.719 fobi.contrib.plugins.form_importers package

48.4.1.1.1.720 Subpackages

48.4.1.1.1.721 fobi.contrib.plugins.form_importers.mailchimp_importer package

48.4.1.1.1.722 Submodules

48.4.1.1.1.723 fobi.contrib.plugins.form_importers.mailchimp_importer.apps module

class fobi.contrib.plugins.form_importers.mailchimp_importer.apps.**Config**(*app_name, app_module*)

Bases: django.apps.config AppConfig

Config.

label = 'fobi_contrib_plugins_form_importers_mailchimp_importer'

name = 'fobi.contrib.plugins.form_importers.mailchimp_importer'

48.4.1.1.1.724 fobi.contrib.plugins.form_importers.mailchimp_importer.base module

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.base.MailChimpImporter(form_entry_cls,
                                                                 form_element_entry_cls,
                                                                 form_properties=None,
                                                                 form_data=None)
```

Bases: *fobi.form_importers.BaseFormImporter*

MailChimp data importer.

extract_field_properties(*field_data*)

Extract field properties.

Handle choices differently as we know what the mailchimp format is.

```
field_properties_mapping = {'choices': 'choices', 'help_text': 'helptext',
                           'initial': 'default', 'label': 'name', 'name': 'tag', 'required': 'req'}
```

```
field_type_prop_name = 'field_type'
```

```
fields_mapping = {'address': 'text', 'date': 'date', 'dropdown': 'select',
                  'email': 'email', 'imageurl': 'url', 'number': 'integer', 'phone': 'text',
                  'radio': 'radio', 'text': 'text', 'url': 'url', 'zip': 'text'}
```

```
name = 'MailChimp'
```

```
position_prop_name = 'order'
```

```
templates = ['mailchimp_importer/0.html', 'mailchimp_importer/1.html']
```

```
uid = 'mailchimp'
```

wizard

alias of *fobi.contrib.plugins.form_importers.mailchimp_importer.views.MailchimpImporterWizardView*

48.4.1.1.1.725 fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers module

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter(form_
                                                                 form_
                                                                 form_
                                                                 form_)
```

Bases: *fobi.form_importers.BaseFormImporter*

MailChimp data importer.

extract_field_properties(*field_data*)

Extract field properties.

Handle choices differently as we know what the mailchimp format is.

```
field_properties_mapping = {'choices': 'choices', 'help_text': 'helptext',
                           'initial': 'default', 'label': 'name', 'name': 'tag', 'required': 'req'}
```

```
field_type_prop_name = 'field_type'
```

```
fields_mapping = {'address': 'text', 'date': 'date', 'dropdown': 'select',
                  'email': 'email', 'imageurl': 'url', 'number': 'integer', 'phone': 'text',
                  'radio': 'radio', 'text': 'text', 'url': 'url', 'zip': 'text'}
```



```

name = 'MailChimp'
position_prop_name = 'order'
templates = ['mailchimp_importer/0.html', 'mailchimp_importer/1.html']
uid = 'mailchimp'
wizard
    alias      of      fobi.contrib.plugins.form_importers.mailchimp_importer.views.
                    MailchimpImporterWizardView

```

48.4.1.1.1.726 fobi.contrib.plugins.form_importers.mailchimp_importer.forms module

```

class fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpAPIKeyForm(data=None,
                                                                                       files=None,
                                                                                       auto_id='id_%s',
                                                                                       pre-
                                                                                       fix=None,
                                                                                       ini-
                                                                                       tial=None,
                                                                                       er-
                                                                                       ror_class=<class
                                                                                       'django.forms.utils.Er-
                                                                                       la-
                                                                                       bel_suffix=None,
                                                                                       empty_permitted=False,
                                                                                       field_order=None,
                                                                                       use_required_attribute=
                                                                                       ren-
                                                                                       derer=None)

```

Bases: django.forms.forms.Form

MailchimpAPIKeyForm.

First form the the wizard. Here users are supposed to provide the API key of their Mailchimp account.

```
base_fields = {'api_key': <django.forms.fields.CharField object>}
```

```
declared_fields = {'api_key': <django.forms.fields.CharField object>}
```

property media

Return all media required to render the widgets on this form.

```

class fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpListIDForm(*args,
                                                                                       **kwargs)

```

Bases: django.forms.forms.Form

MailchimpListIDForm.

Second form of the wizard. Here users are supposed to choose the form they want to import.

```
base_fields = {'list_id': <django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'list_id': <django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

48.4.1.1.1.727 fobi.contrib.plugins.form_importers.mailchimp_importer.views module

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.views.MailchimpImporterWizardView(**kwargs)
    Bases: fobi.wizard.views.views.SessionWizardView
    MailchimpImporterWizardView.
    done(form_list, **kwargs)
        This method must be overridden by a subclass to process to form data after processing all steps.
    form_list = [<class
    'fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpAPIKeyForm'>,
    <class
    'fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpListIDForm'>]
    get_form_kwargs(step)
        Get form kwargs.
```

48.4.1.1.1.728 Module contents

48.4.1.1.1.729 Module contents

48.4.1.1.1.730 Module contents

48.4.1.1.1.731 fobi.contrib.themes package

48.4.1.1.1.732 Subpackages

48.4.1.1.1.733 fobi.contrib.themes.bootstrap3 package

48.4.1.1.1.734 Subpackages

48.4.1.1.1.735 fobi.contrib.themes.bootstrap3.widgets package

48.4.1.1.1.736 Subpackages

48.4.1.1.1.737 fobi.contrib.themes.bootstrap3.widgets.form_elements package

48.4.1.1.1.738 Subpackages

48.4.1.1.1.739 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget package

48.4.1.1.1.740 Submodules

48.4.1.1.1.741 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.apps module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.apps.Config:
```

```
    Bases: django.apps.config AppConfig
```

```
    Config.
```

```
    label =
```

```
    'fobi_contrib_themes_bootstrap3_widgets_form_elements_content_markdown_bootstrap3_widget'
```

```
    name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.'
```

```
    content_markdown_bootstrap3_widget'
```

48.4.1.1.1.742 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.fobi_form module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.fobi_form
    Bases: fobi.contrib.plugins.form\_elements.content.content\_markdown.widgets.
           BaseContentMarkdownPluginWidget
    ContentMarkdownText plugin widget for Bootstrap 3.
    theme_uid = 'bootstrap3'
```

48.4.1.1.1.743 Module contents

48.4.1.1.1.744 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget package

48.4.1.1.1.745 Submodules

48.4.1.1.1.746 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.apps module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.apps.Conf
    Bases: django.apps.config AppConfig
    Config.
    label =
    'fobi_contrib_themes_bootstrap3_widgets_form_elements_content_richtext_bootstrap3_widget'
    name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.
           content_richtext_bootstrap3_widget'
```

48.4.1.1.1.747 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.fobi_form module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.fobi_form
    Bases: fobi.contrib.plugins.form\_elements.content.content\_richtext.widgets.
           BaseContentRichTextPluginWidget
    ContentRichText plugin widget for Bootstrap 3.
    theme_uid = 'bootstrap3'
```

48.4.1.1.1.748 Module contents

48.4.1.1.1.749 `fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget` package

48.4.1.1.1.750 Submodules

48.4.1.1.1.751 `fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps.Config(app_name,
                                                                                               app_module)
    Bases: django.apps.config AppConfig
    Config.
    label =
    'fobi_contrib_themes_bootstrap3_widgets_form_elements_date_bootstrap3_widget'
    name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget'
```

48.4.1.1.1.752 `fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DateBootstrap3Widget
    Bases: fobi.contrib.plugins.form_elements.fields.date.widgets.BaseDatePluginWidget
    Date plugin widget for Bootstrap 3.
    media_css = ['bootstrap3/css/bootstrap-datetimepicker.min.css']
    media_js = ['js/moment-with-locales.js',
                'bootstrap3/js/bootstrap-datetimepicker.min.js',
                'bootstrap3/js/fobi.plugin.date-bootstrap3-widget.js']
    theme_uid = 'bootstrap3'
```

48.4.1.1.1.753 Module contents

48.4.1.1.1.754 `fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget` package

48.4.1.1.1.755 Submodules

48.4.1.1.1.756 `fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps.Config(app_name,
                                                                                               app_module)
    Bases: django.apps.config AppConfig
    Config.
```

```
label =
'fobi_contrib_themes_bootstrap3_widgets_form_elements_datetime_bootstrap3_widget'
name =
'fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget'
```

48.4.1.1.1.757 fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_form_element module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_form_element:
    Bases: fobi.contrib.plugins.form_elements.fields.datetime.widgets.BaseDateTimePluginWidget
```

DateTime plugin widget for Bootstrap 3.

```
media_css = ['bootstrap3/css/bootstrap-datetimepicker.min.css']
```

```
media_js = ['js/moment-with-locales.js',
'bootstrap3/js/bootstrap-datetimepicker.min.js',
'bootstrap3/js/fobi.plugin.datetime-bootstrap3-widget.js']
```

```
theme_uid = 'bootstrap3'
```

48.4.1.1.1.758 Module contents

48.4.1.1.1.759 fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget package

48.4.1.1.1.760 Submodules

48.4.1.1.1.761 fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps.Config(app_name, app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label =
'fobi_contrib_themes_bootstrap3_widgets_form_elements_dummy_bootstrap3_widget'
name =
'fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget'
```

48.4.1.1.1.762 `fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.D
    Bases: fobi.contrib.plugins.form_elements.test.dummy.widgets.BaseDummyPluginWidget

    Dummy plugin widget for Bootstrap 3.

    media_css = []
    media_js = []
    theme_uid = 'bootstrap3'
```

48.4.1.1.1.763 Module contents

48.4.1.1.1.764 `fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget` package

48.4.1.1.1.765 Submodules

48.4.1.1.1.766 `fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps.Config(app_name
    Bases: django.apps.config AppConfig
    Config.

    label =
    'fobi_contrib_themes_bootstrap3_widgets_form_elements_slider_bootstrap3_widget'
    name =
    'fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget'
```

48.4.1.1.1.767 `fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_form_elements` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_form_elements.
    Bases: fobi.contrib.plugins.form_elements.fields.slider.widgets.BaseSliderPluginWidget

    Slider plugin widget for Bootstrap 3.

    media_css = ['bootstrap3/css/bootstrap-slider.min.css',
    'bootstrap3/css/fobi.plugin.slider-bootstrap3-widget.css']
    media_js = ['bootstrap3/js/bootstrap-slider.min.js',
    'bootstrap3/js/fobi.plugin.slider-bootstrap3-widget.js']
    theme_uid = 'bootstrap3'
```

48.4.1.1.1.768 Module contents

48.4.1.1.1.769 Module contents

48.4.1.1.1.770 Module contents

48.4.1.1.1.771 Submodules

48.4.1.1.1.772 fobi.contrib.themes.bootstrap3.apps module

```
class fobi.contrib.themes.bootstrap3.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_themes_bootstrap3'
    name = 'fobi.contrib.themes.bootstrap3'
```

48.4.1.1.1.773 fobi.contrib.themes.bootstrap3.fobi_themes module

```
class fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme(user=None)
    Bases: fobi.base.BaseTheme
    Bootstrap3 theme.
    add_form_element_entry_ajax_template = 'bootstrap3/add_form_element_entry_ajax.html'
    add_form_element_entry_template = 'bootstrap3/add_form_element_entry.html'
    add_form_handler_entry_ajax_template = 'bootstrap3/add_form_handler_entry_ajax.html'
    add_form_handler_entry_template = 'bootstrap3/add_form_handler_entry.html'
    add_form_wizard_handler_entry_ajax_template =
    'bootstrap3/add_form_wizard_handler_entry_ajax.html'
    add_form_wizard_handler_entry_template =
    'bootstrap3/add_form_wizard_handler_entry.html'
    base_template = 'bootstrap3/base.html'
    create_form_entry_ajax_template = 'bootstrap3/create_form_entry_ajax.html'
    create_form_entry_template = 'bootstrap3/create_form_entry.html'
    create_form_wizard_entry_ajax_template =
    'bootstrap3/create_form_wizard_entry_ajax.html'
    create_form_wizard_entry_template = 'bootstrap3/create_form_wizard_entry.html'
    dashboard_template = 'bootstrap3/dashboard.html'
    edit_form_element_entry_ajax_template =
    'bootstrap3/edit_form_element_entry_ajax.html'
    edit_form_element_entry_template = 'bootstrap3/edit_form_element_entry.html'
    edit_form_entry_ajax_template = 'bootstrap3/edit_form_entry_ajax.html'
    edit_form_entry_template = 'bootstrap3/edit_form_entry.html'
```



```

edit_form_handler_entry_ajax_template =
'bootstrap3/edit_form_handler_entry_ajax.html'

edit_form_handler_entry_template = 'bootstrap3/edit_form_handler_entry.html'

edit_form_wizard_entry_ajax_template = 'bootstrap3/edit_form_wizard_entry_ajax.html'

edit_form_wizard_entry_template = 'bootstrap3/edit_form_wizard_entry.html'

edit_form_wizard_handler_entry_ajax_template =
'bootstrap3/edit_form_wizard_handler_entry_ajax.html'

edit_form_wizard_handler_entry_template =
'bootstrap3/edit_form_wizard_handler_entry.html'

embed_form_entry_submitted_ajax_template =
'bootstrap3/embed_form_entry_submitted_ajax.html'

form_ajax = 'bootstrap3/snippets/form_ajax.html'

form_delete_form_entry_option_class = 'glyphicon glyphicon-remove'

form_edit_form_entry_option_class = 'glyphicon glyphicon-edit'

form_element_checkbox_html_class = 'checkbox'

form_element_html_class = 'form-control'

form_entry_submitted_ajax_template = 'bootstrap3/form_entry_submitted_ajax.html'

form_entry_submitted_template = 'bootstrap3/form_entry_submitted.html'

form_importer_ajax_template = 'bootstrap3/form_importer_ajax.html'

form_importer_template = 'bootstrap3/form_importer.html'

form_list_container_class = 'list-inline'

form_non_field_and_hidden_errors_snippet_template =
'bootstrap3/snippets/form_non_field_and_hidden_errors_snippet.html'

form_properties_snippet_template_name =
'bootstrap3/snippets/form_properties_snippet.html'

form_snippet_template_name = 'bootstrap3/snippets/form_snippet.html'

form_view_form_entry_option_class = 'glyphicon glyphicon-list'

form_wizard_ajax = 'bootstrap3/snippets/form_wizard_ajax.html'

form_wizard_properties_snippet_template_name =
'bootstrap3/snippets/form_wizard_properties_snippet.html'

form_wizard_snippet_template_name = 'bootstrap3/snippets/form_wizard_snippet.html'

form_wizard_template = 'bootstrap3/snippets/form_wizard.html'

form_wizards_dashboard_template = 'bootstrap3/form_wizards_dashboard.html'

forms_list_template = 'bootstrap3/forms_list.html'

master_base_template = 'bootstrap3/_base.html'

media_css = ('bootstrap3/css/bootstrap.css',
'bootstrap3/css/bootstrap3_fobi_extras.css', 'css/fobi.core.css')

```

```
media_js = ('js/jquery-1.10.2.min.js',
'jquery-ui/js/jquery-ui-1.10.4.custom.min.js', 'bootstrap3/js/bootstrap.min.js',
'js/jquery.slugify.js', 'js/fobi.core.js',
'bootstrap3/js/bootstrap3_fobi_extras.js')

messages_snippet_template_name = 'bootstrap3/snippets/messages_snippet.html'
name = 'Bootstrap 3'
uid = 'bootstrap3'
view_embed_form_entry_ajax_template = 'bootstrap3/view_embed_form_entry_ajax.html'
view_form_entry_ajax_template = 'bootstrap3/view_form_entry_ajax.html'
view_form_entry_template = 'bootstrap3/view_form_entry.html'
view_form_wizard_entry_ajax_template = 'bootstrap3/view_form_wizard_entry_ajax.html'
view_form_wizard_entry_template = 'bootstrap3/view_form_wizard_entry.html'
```

48.4.1.1.1.774 Module contents

48.4.1.1.1.775 fobi.contrib.themes.djangocms_admin_style_theme package

48.4.1.1.1.776 Subpackages

48.4.1.1.1.777 fobi.contrib.themes.djangocms_admin_style_theme.widgets package

48.4.1.1.1.778 Subpackages

48.4.1.1.1.779 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements package

48.4.1.1.1.780 Subpackages

48.4.1.1.1.781 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_style_theme module

48.4.1.1.1.782 Submodules

48.4.1.1.1.783 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_style_theme.widget module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_style_theme.widget

    Bases: django.apps.config.AppConfig
    Config.
    label =
'fobi_contrib_themes.djangocms_admin_style_theme_widgets_form_elements_content_richtext_admin_style_theme'
    name = 'fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_style_widget'
```

48.4.1.1.1.784 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_style_theme module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_style_theme(
    Bases: fobi.contrib.plugins.form_elements.content.content_richtext.widgets.BaseContentRichTextPluginWidget
    ContentRichText plugin widget for.djangocms_admin_style theme.
    theme_uid = 'simple'
```

48.4.1.1.1.785 Module contents

48.4.1.1.1.786 Module contents

48.4.1.1.1.787 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers package

48.4.1.1.1.788 Subpackages

48.4.1.1.1.789 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store package

48.4.1.1.1.790 Submodules

48.4.1.1.1.791 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps.Config(app_name, app_label,
    Bases: django.apps.config AppConfig
    Config.
    label =
    'fobi_contrib_themes.djangocms_admin_style_theme_widgets_form_handlers_db_store'
    name =
    'fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store'
```

48.4.1.1.1.792 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements(
    Bases: fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget
    DbStore plugin widget for.djangocms_admin_style_theme theme.
    export_entries_icon_class = ''
    theme_uid = 'djangocms_admin_style_theme'
    view_entries_icon_class = ''
```

48.4.1.1.1.793 Module contents

48.4.1.1.1.794 Module contents

48.4.1.1.1.795 Module contents

48.4.1.1.1.796 Submodules

48.4.1.1.1.797 fobi.contrib.themes.djangocms_admin_style_theme.apps module

```
class fobi.contrib.themes.djangocms_admin_style_theme.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_themes.djangocms_admin_style_theme'
    name = 'fobi.contrib.themes.djangocms_admin_style_theme'
```

48.4.1.1.1.798 fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes module

```
class fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme(user=None)
    Bases: fobi.base.BaseTheme
    A theme that has a native.djangocms-admin-style style.
    add_form_element_entry_ajax_template =
    'djangocms_admin_style_theme/add_form_element_entry_ajax.html'
    add_form_element_entry_template =
    'djangocms_admin_style_theme/add_form_element_entry.html'
    add_form_handler_entry_ajax_template =
    'djangocms_admin_style_theme/add_form_handler_entry_ajax.html'
    add_form_handler_entry_template =
    'djangocms_admin_style_theme/add_form_handler_entry.html'
    base_edit_template = 'djangocms_admin_style_theme/base_edit.html'
    base_template = 'djangocms_admin_style_theme/base.html'
    base_view_template = 'djangocms_admin_style_theme/base_view.html'
    create_form_entry_ajax_template =
    'djangocms_admin_style_theme/create_form_entry_ajax.html'
    create_form_entry_template = 'djangocms_admin_style_theme/create_form_entry.html'
    dashboard_template = 'djangocms_admin_style_theme/dashboard.html'
    edit_form_element_entry_ajax_template =
    'djangocms_admin_style_theme/edit_form_element_entry_ajax.html'
    edit_form_element_entry_template =
    'djangocms_admin_style_theme/edit_form_element_entry.html'
    edit_form_entry_ajax_template =
    'djangocms_admin_style_theme/edit_form_entry_ajax.html'
```

```
classmethod edit_form_entry_edit_option_html()
```

For adding the edit link to edit form entry view.

Return str

```
classmethod edit_form_entry_help_text_extra()
```

For adding the edit link to edit form entry view.

Return str

```
edit_form_entry_template = 'djangocms_admin_style_theme/edit_form_entry.html'
```

```
edit_form_handler_entry_ajax_template =  
'djangocms_admin_style_theme/edit_form_handler_entry_ajax.html'
```

```
edit_form_handler_entry_template =  
'djangocms_admin_style_theme/edit_form_handler_entry.html'
```

```
form_ajax = 'djangocms_admin_style_theme/snippets/form_ajax.html'
```

```
form_delete_form_entry_option_class = 'deletelink'
```

```
form_edit_ajax = 'djangocms_admin_style_theme/snippets/form_edit_ajax.html'
```

```
form_edit_form_entry_option_class = 'edit'
```

```
form_edit_snippet_template_name =  
'djangocms_admin_style_theme/snippets/form_edit_snippet.html'
```

```
form_element_checkbox_html_class = 'checkbox'
```

```
form_element_html_class = 'vTextField'
```

```
form_entry_submitted_ajax_template =  
'djangocms_admin_style_theme/form_entry_submitted_ajax.html'
```

```
form_entry_submitted_template =  
'djangocms_admin_style_theme/form_entry_submitted.html'
```

```
form_list_container_class = 'list-inline'
```

```
form_properties_snippet_template_name =  
'djangocms_admin_style_theme/snippets/form_properties_snippet.html'
```

```
form_radio_element_html_class = 'radiolist'
```

```
form_snippet_template_name =  
'djangocms_admin_style_theme/snippets/form_snippet.html'
```

```
form_view_form_entry_option_class = 'viewlink'
```

```
form_view_snippet_template_name =  
'djangocms_admin_style_theme/snippets/form_view_snippet.html'
```

```
forms_list_template = 'djangocms_admin_style_theme/forms_list.html'
```

```
import_form_entry_ajax_template =  
'djangocms_admin_style_theme/import_form_entry_ajax.html'
```

```
import_form_entry_template = 'djangocms_admin_style_theme/import_form_entry.html'
```

```
master_base_template = 'djangocms_admin_style_theme/_base.html'
```

```
media_css = ('djangocms_admin_style_theme/css/fobi.djangocms_admin_style_theme.css',  
'jquery-ui/css/smoothness/jquery-ui-1.10.3.custom.min.css')
```

```
media_js = ('js/jquery-1.10.2.min.js',
'jquery-ui/js/jquery-ui-1.10.4.custom.min.js', 'js/jquery.slugify.js',
'js/fobi.core.js')
messages_snippet_template_name =
'djangocms_admin_style_theme/snippets/messages_snippet.html'
name = 'DjangoCMS admin style'
uid = 'djangocms_admin_style_theme'
view_form_entry_ajax_template =
'djangocms_admin_style_theme/view_form_entry_ajax.html'
view_form_entry_template = 'djangocms_admin_style_theme/view_form_entry.html'
```

48.4.1.1.1.799 Module contents

48.4.1.1.1.800 fobi.contrib.themes.foundation5 package

48.4.1.1.1.801 Subpackages

48.4.1.1.1.802 fobi.contrib.themes.foundation5.widgets package

48.4.1.1.1.803 Subpackages

48.4.1.1.1.804 fobi.contrib.themes.foundation5.widgets.form_elements package

48.4.1.1.1.805 Subpackages

48.4.1.1.1.806 fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget package

48.4.1.1.1.807 Submodules

48.4.1.1.1.808 fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget.apps module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget.apps.Config
```

```
    Bases: django.apps.config AppConfig
```

```
    Config.
```

```
    label =
```

```
    'fobi_contrib_themes_foundation5_widgets_form_elements_content_richtext_foundation5_widget'
```

```
    name = 'fobi.contrib.themes.foundation5.widgets.form_elements.
```

```
    content_richtext_foundation5_widget'
```

48.4.1.1.1.809 `fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget.fobi_fo` module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget.fobi_fo
    Bases: fobi.contrib.plugins.form_elements.content.content_richtext.widgets.
           BaseContentRichTextPluginWidget
    Content rich-text plugin widget for Foundation 5.
    theme_uid = 'foundation5'
```

48.4.1.1.1.810 Module contents

48.4.1.1.1.811 `fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget` package

48.4.1.1.1.812 Submodules

48.4.1.1.1.813 `fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps` module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps.Config(app_name
                                                    app_mod

    Bases: django.apps.config AppConfig
    Config.
    label =
    'fobi_contrib_themes_foundation5_widgets_form_elements_date_foundation5_widget'
    name =
    'fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget'
```

48.4.1.1.1.814 `fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form_eleme` module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form_elements.
    Bases: fobi.contrib.plugins.form_elements.fields.date.widgets.BaseDatePluginWidget
    Date plugin widget for Foundation 5.
    media_css = ['foundation5/css/foundation-datepicker.css']
    media_js = ['js/moment-with-locales.js', 'foundation5/js/foundation-datepicker.js',
    'foundation5/js/fobi.plugin.date-foundation5-widget.js']
    theme_uid = 'foundation5'
```

48.4.1.1.1.815 Module contents

48.4.1.1.1.816 `fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget` package

48.4.1.1.1.817 Submodules

48.4.1.1.1.818 `fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps` module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps.Config(app_name='fobi',
app_label='fobi')
    Bases: django.apps.config AppConfig
    Config.
    label =
    'fobi_contrib_themes_foundation5_widgets_form_elements_datetime_foundation5_widget'
    name =
    'fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget'
```

48.4.1.1.1.819 `fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.fobi_form_elements` module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.fobi_form_elements(BaseDateTimePluginWidget)
    Bases: fobi.contrib.plugins.form_elements.fields.datetime.widgets.BaseDateTimePluginWidget
    BaseDateTimePluginWidget
    DateTime plugin widget for Foundation 5.
    media_css = ['foundation5/css/foundation-datetimepicker.css']
    media_js = ['js/moment-with-locales.js',
    'foundation5/js/foundation-datetimepicker.js',
    'foundation5/js/fobi.plugin.datetime-foundation5-widget.js']
    theme_uid = 'foundation5'
```

48.4.1.1.1.820 Module contents

48.4.1.1.1.821 `fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget` package

48.4.1.1.1.822 Submodules

48.4.1.1.1.823 `fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps` module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps.Config(app_name='fobi',
app_label='fobi')
    Bases: django.apps.config AppConfig
    Config.
```



```
label =
'fobi_contrib_themes_foundation5_widgets_form_elements_dummy_foundation5_widget'
name =
'fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget'
```

48.4.1.1.1.824 fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_elements module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_elements
    Bases: fobi.contrib.plugins.form_elements.test.dummy.widgets.BaseDummyPluginWidget
    Dummy plugin widget for Foundation 5.
    media_css = []
    media_js = []
    theme_uid = 'foundation5'
```

48.4.1.1.1.825 Module contents

48.4.1.1.1.826 Module contents

48.4.1.1.1.827 fobi.contrib.themes.foundation5.widgets.form_handlers package

48.4.1.1.1.828 Subpackages

48.4.1.1.1.829 fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget package

48.4.1.1.1.830 Submodules

48.4.1.1.1.831 fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps module

```
class fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps.Config(app_
    Bases: django.apps.config AppConfig
    Config.
    label =
'fobi_contrib_themes_foundation5_widgets_form_handlers_db_store_foundation5_widget'
    name =
'fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget'
```

48.4.1.1.1.832 `fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_element` module

`class fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_element`
 Bases: `fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget`

DbStore plugin widget for Foundation 5.

`export_entries_icon_class = 'fi-page-export'`

`theme_uid = 'foundation5'`

`view_entries_icon_class = 'fi-list'`

`view_saved_form_data_entries_template_name =`
`'db_store_foundation5_widget/view_saved_form_data_entries.html'`

48.4.1.1.1.833 Module contents

48.4.1.1.1.834 Module contents

48.4.1.1.1.835 Module contents

48.4.1.1.1.836 Submodules

48.4.1.1.1.837 `fobi.contrib.themes.foundation5.apps` module

`class fobi.contrib.themes.foundation5.apps.Config(app_name, app_module)`
 Bases: `django.apps.config.AppConfig`

Config.

`label = 'fobi_contrib_themes_foundation5'`

`name = 'fobi.contrib.themes.foundation5'`

48.4.1.1.1.838 `fobi.contrib.themes.foundation5.fobi_themes` module

`class fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme(user=None)`
 Bases: `fobi.base.BaseTheme`

Foundation5 theme.

Based on the “Workspace” example of the Foundation 5. Click [here](#) for more.

`add_form_element_entry_ajax_template =`
`'foundation5/add_form_element_entry_ajax.html'`

`add_form_element_entry_template = 'foundation5/add_form_element_entry.html'`

`add_form_handler_entry_ajax_template =`
`'foundation5/add_form_handler_entry_ajax.html'`

`add_form_handler_entry_template = 'foundation5/add_form_handler_entry.html'`

`base_template = 'foundation5/base.html'`

`create_form_entry_ajax_template = 'foundation5/create_form_entry_ajax.html'`

```

create_form_entry_template = 'foundation5/create_form_entry.html'
create_form_wizard_entry_ajax_template =
'foundation5/create_form_wizard_entry_ajax.html'
create_form_wizard_entry_template = 'foundation5/create_form_wizard_entry.html'
dashboard_template = 'foundation5/dashboard.html'
edit_form_element_entry_ajax_template =
'foundation5/edit_form_element_entry_ajax.html'
edit_form_element_entry_template = 'foundation5/edit_form_element_entry.html'
edit_form_entry_ajax_template = 'foundation5/edit_form_entry_ajax.html'
edit_form_entry_template = 'foundation5/edit_form_entry.html'
edit_form_handler_entry_ajax_template =
'foundation5/edit_form_handler_entry_ajax.html'
edit_form_handler_entry_template = 'foundation5/edit_form_handler_entry.html'
edit_form_wizard_entry_ajax_template =
'foundation5/edit_form_wizard_entry_ajax.html'
edit_form_wizard_entry_template = 'foundation5/edit_form_wizard_entry.html'
form_ajax = 'foundation5/snippets/form_ajax.html'
form_delete_form_entry_option_class = 'fi-page-delete'
form_edit_form_entry_option_class = 'fi-page-edit'
form_element_checkbox_html_class = 'checkbox'
form_element_html_class = 'form-control'
form_entry_submitted_ajax_template = 'foundation5/form_entry_submitted_ajax.html'
form_entry_submitted_template = 'foundation5/form_entry_submitted.html'
form_list_container_class = 'inline-list'
form_non_field_and_hidden_errors_snippet_template =
'foundation5/snippets/form_non_field_and_hidden_errors_snippet.html'
form_properties_snippet_template_name =
'foundation5/snippets/form_properties_snippet.html'
form_snippet_template_name = 'foundation5/snippets/form_snippet.html'
form_view_form_entry_option_class = 'fi-list'
form_wizard_ajax = 'foundation5/snippets/form_wizard_ajax.html'
form_wizard_properties_snippet_template_name =
'foundation5/snippets/form_wizard_properties_snippet.html'
form_wizard_snippet_template_name = 'foundation5/snippets/form_wizard_snippet.html'
form_wizards_dashboard_template = 'foundation5/form_wizards_dashboard.html'
forms_list_template = 'foundation5/forms_list.html'
import_form_entry_ajax_template = 'foundation5/import_form_entry_ajax.html'
import_form_entry_template = 'foundation5/import_form_entry.html'

```

```
master_base_template = 'foundation5/_base.html'
media_css = ('foundation5/css/foundation.min.css',
'foundation5/css/foundation_fobi_extras.css',
'foundation5/icons/3/icons/foundation-icons.css')
media_js = ('foundation5/js/vendor/modernizr.js', 'foundation5/js/vendor/jquery.js',
'jquery-ui/js/jquery-ui-1.10.4.custom.min.js', 'foundation5/js/foundation.min.js',
'js/fobi.core.js', 'js/jquery.slugify.js',
'foundation5/js/foundation5_fobi_extras.js')
messages_snippet_template_name = 'foundation5/snippets/messages_snippet.html'
name = 'Foundation 5'
uid = 'foundation5'
view_form_entry_ajax_template = 'foundation5/view_form_entry_ajax.html'
view_form_entry_template = 'foundation5/view_form_entry.html'
view_form_wizard_entry_ajax_template =
'foundation5/view_form_wizard_entry_ajax.html'
view_form_wizard_entry_template = 'foundation5/view_form_wizard_entry.html'
```

48.4.1.1.1839 Module contents

48.4.1.1.1840 fobi.contrib.themes.simple package

48.4.1.1.1841 Subpackages

48.4.1.1.1842 fobi.contrib.themes.simple.widgets package

48.4.1.1.1843 Subpackages

48.4.1.1.1844 fobi.contrib.themes.simple.widgets.form_elements package

48.4.1.1.1845 Subpackages

48.4.1.1.1846 fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget package

48.4.1.1.1847 Submodules

48.4.1.1.1848 fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.apps module

```
class fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.apps.Config(app_name, app_label)
    Bases: django.apps.config AppConfig
    Config.
```

```
label =
'fobi_contrib_themes_simple_widgets_form_elements_content_richtext_simple_widget'
name =
'fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget'
```

48.4.1.1.1.849 fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi_form_element module

```
class fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi_form_element:
    Bases: fobi.contrib.plugins.form_elements.content.content_richtext.widgets.BaseContentRichTextPluginWidget
    ContentRichText plugin widget for Simple theme.
    theme_uid = 'simple'
```

48.4.1.1.1.850 Module contents

48.4.1.1.1.851 Module contents

48.4.1.1.1.852 fobi.contrib.themes.simple.widgets.form_handlers package

48.4.1.1.1.853 Subpackages

48.4.1.1.1.854 fobi.contrib.themes.simple.widgets.form_handlers.db_store package

48.4.1.1.1.855 Submodules

48.4.1.1.1.856 fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps module

```
class fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_simple_widgets_form_handlers_db_store'
    name = 'fobi.contrib.themes.simple.widgets.form_handlers.db_store'
```

48.4.1.1.1.857 fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements module

```
class fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements.DbStorePluginWidget:
    Bases: fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget
    DbStore plugin widget for Simple theme.
    export_entries_icon_class = ''
    theme_uid = 'simple'
    view_entries_icon_class = ''
```

48.4.1.1.1.858 Module contents

48.4.1.1.1.859 Module contents

48.4.1.1.1.860 Module contents

48.4.1.1.1.861 Submodules

48.4.1.1.1.862 fobi.contrib.themes.simple.apps module

```
class fobi.contrib.themes.simple.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_themes_simple'
    name = 'fobi.contrib.themes.simple'
```

48.4.1.1.1.863 fobi.contrib.themes.simple.fobi_themes module

```
class fobi.contrib.themes.simple.fobi_themes.SimpleTheme(user=None)
    Bases: fobi.base.BaseTheme
    Simple theme that has a native Django style.
    add_form_element_entry_ajax_template = 'simple/add_form_element_entry_ajax.html'
    add_form_element_entry_template = 'simple/add_form_element_entry.html'
    add_form_handler_entry_ajax_template = 'simple/add_form_handler_entry_ajax.html'
    add_form_handler_entry_template = 'simple/add_form_handler_entry.html'
    base_edit_template = 'simple/base_edit.html'
    base_template = 'simple/base.html'
    base_view_template = 'simple/base_view.html'
    create_form_entry_ajax_template = 'simple/create_form_entry_ajax.html'
    create_form_entry_template = 'simple/create_form_entry.html'
    create_form_wizard_entry_ajax_template = 'simple/create_form_wizard_entry_ajax.html'
    create_form_wizard_entry_template = 'simple/create_form_wizard_entry.html'
    dashboard_template = 'simple/dashboard.html'
    edit_form_element_entry_ajax_template = 'simple/edit_form_element_entry_ajax.html'
    edit_form_element_entry_template = 'simple/edit_form_element_entry.html'
    edit_form_entry_ajax_template = 'simple/edit_form_entry_ajax.html'
    edit_form_entry_template = 'simple/edit_form_entry.html'
    edit_form_handler_entry_ajax_template = 'simple/edit_form_handler_entry_ajax.html'
    edit_form_handler_entry_template = 'simple/edit_form_handler_entry.html'
    edit_form_wizard_entry_ajax_template = 'simple/edit_form_wizard_entry_ajax.html'
```

```

edit_form_wizard_entry_template = 'simple/edit_form_wizard_entry.html'
form_ajax = 'simple/snippets/form_ajax.html'
form_delete_form_entry_option_class = 'glyphicon glyphicon-remove'
form_edit_ajax = 'simple/snippets/form_edit_ajax.html'
form_edit_form_entry_option_class = 'glyphicon glyphicon-edit'
form_edit_snippet_template_name = 'simple/snippets/form_edit_snippet.html'
form_element_checkbox_html_class = 'checkbox'
form_element_html_class = 'vTextField'
form_entry_submitted_ajax_template = 'simple/form_entry_submitted_ajax.html'
form_entry_submitted_template = 'simple/form_entry_submitted.html'
form_list_container_class = 'list-inline'
form_properties_snippet_template_name =
'simple/snippets/form_properties_snippet.html'
form_radio_element_html_class = 'radiolist'
form_snippet_template_name = 'simple/snippets/form_snippet.html'
form_view_form_entry_option_class = 'glyphicon glyphicon-list'
form_view_snippet_template_name = 'simple/snippets/form_view_snippet.html'
form_wizard_properties_snippet_template_name =
'simple/snippets/form_wizard_properties_snippet.html'
form_wizard_snippet_template_name = 'simple/snippets/form_wizard_snippet.html'
forms_list_template = 'simple/forms_list.html'
import_form_entry_ajax_template = 'simple/import_form_entry_ajax.html'
import_form_entry_template = 'simple/import_form_entry.html'
master_base_template = 'simple/_base.html'
media_css = ('simple/css/fobi.simple.css',
'jquery-ui/css/django-admin-theme/jquery-ui-1.10.4.custom.min.css')
media_js = ('js/jquery-1.10.2.min.js',
'jquery-ui/js/jquery-ui-1.10.4.custom.min.js', 'js/jquery.slugify.js',
'js/fobi.core.js')
messages_snippet_template_name = 'simple/snippets/messages_snippet.html'
name = 'Django admin style'
uid = 'simple'
view_form_entry_ajax_template = 'simple/view_form_entry_ajax.html'
view_form_entry_template = 'simple/view_form_entry.html'
view_form_wizard_entry_ajax_template = 'simple/view_form_wizard_entry_ajax.html'
view_form_wizard_entry_template = 'simple/view_form_wizard_entry.html'

```

48.4.1.1.1.864 Module contents

48.4.1.1.1.865 Module contents

48.4.1.1.2 Module contents

48.4.1.2 fobi.integration package

48.4.1.2.1 Submodules

48.4.1.2.2 fobi.integration.helpers module

`fobi.integration.helpers.get_template_choices(source, choices, theme_specific_choices_key)`

Get the template choices.

It's possible to provide theme templates per theme or just per project.

Parameters

- **source** (*str*) – Example value 'feincms_integration'.
- **or list choices** (*tuple*) –
- **theme_specific_choices_key** (*str*) –

Return list

48.4.1.2.3 fobi.integration.processors module

`class fobi.integration.processors.IntegrationProcessor`

Bases: object

Generic integration processor.

Parameters

- **form_sent_get_param** (*str*) –
- **can_redirect** (*bool*) – If set to True, if not authenticated an attempt to redirect user to a login page would be made. Otherwise, a message about authentication would be generated instead (in place of the form). Some content management systems, like Django-CMS, aren't able to redirect on plugin level. For those systems, the value of `can_redirect` should be set to False.
- **login_required_template_name** (*str*) – Template to be used for rendering the login required message. This is only important when `login_required_redirect` is set to False.

`can_redirect = True`

`form_sent_get_param = 'sent'`

`get_context_data(request, instance, **kwargs)`

Get context data.

`get_form_template_name(request, instance)`

Get form template name.

`get_login_required_template_name(request, instance)`

Get login required template name.

get_process_form_redirect_url(*request*, *instance*)

Get process form redirect URL (success).

Parameters

- **request** (*django.http.HttpRequest*) –
- **instance** (*fobi.models.FormEntry*) –

Return str

get_success_page_template_name(*request*, *instance*)

Get succes page template name.

integration_check(*instance*)

Integration check.

Performs a simple check to identify whether the model instance has been implemented according to the expectations.

login_required_template_name = 'fobi/integration/login_required.html'

48.4.1.2.4 Module contents

48.4.1.3 fobi.management package

48.4.1.3.1 Subpackages

48.4.1.3.1.1 fobi.management.commands package

48.4.1.3.1.2 Submodules

48.4.1.3.1.3 fobi.management.commands.fobi_find_broken_entries module

```
class fobi.management.commands.fobi_find_broken_entries.Command(stdout=None, stderr=None,  
                                                                no_color=False,  
                                                                force_color=False)
```

Bases: *django.core.management.base.BaseCommand*

Find the broken plugin records in the database:

- *fobi.models.FormElementEntry*
- *fobi.models.FormHandlerEntry*

handle(**args*, ***options*)

Handle.

48.4.1.3.1.4 fobi.management.commands.fobi_sync_plugins module

```
class fobi.management.commands.fobi_sync_plugins.Command(stdout=None, stderr=None,
                                                    no_color=False, force_color=False)

    Bases: django.core.management.base.BaseCommand

    Adds the missing plugins to database.

    This command shall be ran every time a developer adds a new plugin. The following plugins are affected:

    • fobi.models.FormElementPlugin

    • fobi.models.FormHandlerPlugin

    add_arguments(parser)
        Entry point for subclassed commands to add custom arguments.

    handle(*args, **options)
        Handle.
```

48.4.1.3.1.5 fobi.management.commands.fobi_update_plugin_data module

```
class fobi.management.commands.fobi_update_plugin_data.Command(stdout=None, stderr=None,
                                                    no_color=False,
                                                    force_color=False)

    Bases: django.core.management.base.BaseCommand

    Updates the plugin data for all entries of all users.

    Rules for update are specified in the plugin itself.

    This command shall be ran if significant changes have been made to the system for which the data shall be
    updated.

    handle(*args, **options)
        Handle.
```

48.4.1.3.1.6 Module contents

48.4.1.3.2 Module contents

48.4.1.4 fobi.migrations package

48.4.1.4.1 Submodules

48.4.1.4.2 fobi.migrations.0001_initial module

```
class fobi.migrations.0001_initial.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('auth', '__first__'), ('auth', '0006_require_contenttypes_0002')]
```

```

operations = [<CreateModel name='FormElement', fields=[('id',
<django.db.models.fields.AutoField>), ('plugin_uid',
<django.db.models.fields.CharField>), ('groups',
<django.db.models.fields.related.ManyToManyField>), ('users',
<django.db.models.fields.related.ManyToManyField>)], options={'abstract': False,
'verbose_name': 'Form element plugin', 'verbose_name_plural': 'Form element
plugins'}>, <CreateModel name='FormElementEntry', fields=[('id',
<django.db.models.fields.AutoField>), ('plugin_data',
<django.db.models.fields.TextField>), ('plugin_uid',
<django.db.models.fields.CharField>), ('position',
<django.db.models.fields.PositiveIntegerField>)], options={'ordering':
['position'], 'abstract': False, 'verbose_name': 'Form element entry',
'verbose_name_plural': 'Form element entries'}>, <CreateModel name='FormEntry',
fields=[('id', <django.db.models.fields.AutoField>), ('name',
<django.db.models.fields.CharField>), ('slug', <autoslug.fields.AutoSlugField>),
('is_public', <django.db.models.fields.BooleanField>), ('is_cloneable',
<django.db.models.fields.BooleanField>), ('position',
<django.db.models.fields.PositiveIntegerField>), ('success_page_title',
<django.db.models.fields.CharField>), ('success_page_message',
<django.db.models.fields.TextField>), ('action',
<django.db.models.fields.CharField>)], options={'verbose_name': 'Form entry',
'verbose_name_plural': 'Form entries'}>, <CreateModel name='FormFieldsetEntry',
fields=[('id', <django.db.models.fields.AutoField>), ('name',
<django.db.models.fields.CharField>), ('is_repeatable',
<django.db.models.fields.BooleanField>), ('form_entry',
<django.db.models.fields.related.ForeignKey>)], options={'verbose_name': 'Form
fieldset entry', 'verbose_name_plural': 'Form fieldset entries'}>, <CreateModel
name='FormHandler', fields=[('id', <django.db.models.fields.AutoField>),
('plugin_uid', <django.db.models.fields.CharField>), ('groups',
<django.db.models.fields.related.ManyToManyField>), ('users',
<django.db.models.fields.related.ManyToManyField>)], options={'abstract': False,
'verbose_name': 'Form handler plugin', 'verbose_name_plural': 'Form handler
plugins'}>, <CreateModel name='FormHandlerEntry', fields=[('id',
<django.db.models.fields.AutoField>), ('plugin_data',
<django.db.models.fields.TextField>), ('plugin_uid',
<django.db.models.fields.CharField>), ('form_entry',
<django.db.models.fields.related.ForeignKey>)], options={'abstract': False,
'verbose_name': 'Form handler entry', 'verbose_name_plural': 'Form handler
entries'}>, <CreateModel name='FormWizardEntry', fields=[('id',
<django.db.models.fields.AutoField>), ('name', <django.db.models.fields.CharField>),
('slug', <autoslug.fields.AutoSlugField>), ('is_public',
<django.db.models.fields.BooleanField>), ('is_cloneable',
<django.db.models.fields.BooleanField>), ('user',
<django.db.models.fields.related.ForeignKey>)], options={'verbose_name': 'Form
wizard entry', 'verbose_name_plural': 'Form wizard entries'}>, <AddField
model_name='formentry', name='form_wizard_entry',
field=<django.db.models.fields.related.ForeignKey>>, <AddField
model_name='formentry', name='user',
field=<django.db.models.fields.related.ForeignKey>>, <AddField
model_name='formelemententry', name='form_entry',
field=<django.db.models.fields.related.ForeignKey>>, <AddField
model_name='formelemententry', name='form_fieldset_entry',
field=<django.db.models.fields.related.ForeignKey>>, <AlterUniqueTogether
name='formwizardentry', unique_together={('user', 'slug'), ('user', 'name')}>,
<AlterUniqueTogether name='formfieldsetentry', unique_together={('form_entry',
'name')}>>, <AlterUniqueTogether name='formentry', unique_together={('user', 'slug'),
('plugin_data', 'name')}>>]

```

48.4.1.4.3 fobi.migrations.0002_auto_20150912_1744 module

```
class fobi.migrations.0002_auto_20150912_1744.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0001_initial')]

    operations = [<AddField model_name='formentry', name='created',
field=<django.db.models.fields.DateTimeField>>, <AddField model_name='formentry',
name='updated', field=<django.db.models.fields.DateTimeField>>]
```

48.4.1.4.4 fobi.migrations.0003_auto_20160517_1005 module

```
class fobi.migrations.0003_auto_20160517_1005.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0002_auto_20150912_1744')]

    operations = [<AlterField model_name='formelement', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='formelemententry', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField model_name='formentry',
name='slug', field=<autoslug.fields.AutoSlugField>>, <AlterField
model_name='formhandler', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='formhandlerentry', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='formwizardentry', name='slug', field=<autoslug.fields.AutoSlugField>>]
```

48.4.1.4.5 fobi.migrations.0004_auto_20160906_1513 module

```
class fobi.migrations.0004_auto_20160906_1513.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0003_auto_20160517_1005')]

    operations = [<AlterField model_name='formelement', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='formelemententry', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField model_name='formhandler',
name='plugin_uid', field=<django.db.models.fields.CharField>>, <AlterField
model_name='formhandlerentry', name='plugin_uid',
field=<django.db.models.fields.CharField>>]
```

48.4.1.4.6 fobi.migrations.0005_auto_20160908_1457 module

```
class fobi.migrations.0005_auto_20160908_1457.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0004_auto_20160906_1513')]

    operations = [<AlterField model_name='formelement', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='formelemententry', name='plugin_uid',
field=<django.db.models.fields.CharField>>]
```

48.4.1.4.7 fobi.migrations.0006_auto_20160911_1549 module

```
class fobi.migrations.0006_auto_20160911_1549.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0005_auto_20160908_1457')]

    operations = [<AlterField model_name='formhandler', name='plugin_uid',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='formhandlerentry', name='plugin_uid',
field=<django.db.models.fields.CharField>>]
```

48.4.1.4.8 fobi.migrations.0007_auto_20160926_1652 module

```
class fobi.migrations.0007_auto_20160926_1652.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0006_auto_20160911_1549')]

    operations = [<CreateModel name='FormWizardFormEntry', fields=[('id',
<django.db.models.fields.AutoField>), ('position',
<django.db.models.fields.PositiveIntegerField>)], options={'ordering':
['position'], 'abstract': False, 'verbose_name': 'Form wizard form entry',
'verbose_name_plural': 'Form wizard form entries'}>, <RemoveField
model_name='formentry', name='form_wizard_entry'>, <RemoveField
model_name='formentry', name='position'>, <AddField model_name='formwizardentry',
name='created', field=<django.db.models.fields.DateTimeField>>, <AddField
model_name='formwizardentry', name='success_page_message',
field=<django.db.models.fields.TextField>>, <AddField model_name='formwizardentry',
name='success_page_title', field=<django.db.models.fields.CharField>>, <AddField
model_name='formwizardentry', name='updated',
field=<django.db.models.fields.DateTimeField>>, <AddField
model_name='formwizardformentry', name='form_entry',
field=<django.db.models.fields.related.ForeignKey>>, <AddField
model_name='formwizardformentry', name='form_wizard_entry',
field=<django.db.models.fields.related.ForeignKey>>, <AlterUniqueTogether
name='formwizardformentry', unique_together=[('form_wizard_entry', 'form_entry')]>]
```

48.4.1.4.9 fobi.migrations.0008_formwizardhandlerentry module

```
class fobi.migrations.0008_formwizardhandlerentry.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0007_auto_20160926_1652')]

    operations = [<CreateModel name='FormWizardHandlerEntry', fields=[('id',
<django.db.models.fields.AutoField>), ('plugin_data',
<django.db.models.fields.TextField>), ('plugin_uid',
<django.db.models.fields.CharField>), ('form_wizard_entry',
<django.db.models.fields.related.ForeignKey>)], options={'abstract': False,
'verbose_name': 'Form wizard handler entry', 'verbose_name_plural': 'Form wizard
handler entries'}>]
```

48.4.1.4.10 fobi.migrations.0009_formwizardentry_wizard_type module

```
class fobi.migrations.0009_formwizardentry_wizard_type.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0008_formwizardhandlerentry')]

    operations = [<AddField model_name='formwizardentry', name='wizard_type',
field=<django.db.models.fields.CharField>>]
```

48.4.1.4.11 fobi.migrations.0010_formwizardhandler module

```
class fobi.migrations.0010_formwizardhandler.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('auth', '0007_alter_validators_add_error_messages'), ('auth',
'__first__'), ('fobi', '0009_formwizardentry_wizard_type')]

    operations = [<CreateModel name='FormWizardHandler', fields=[('id',
<django.db.models.fields.AutoField>), ('plugin_uid',
<django.db.models.fields.CharField>), ('groups',
<django.db.models.fields.related.ManyToManyField>), ('users',
<django.db.models.fields.related.ManyToManyField>)], options={'abstract': False,
'verbose_name': 'Form wizard handler plugin', 'verbose_name_plural': 'Form wizard
handler plugins'}>]
```

48.4.1.4.12 fobi.migrations.0011_formentry_title module

```
class fobi.migrations.0011_formentry_title.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0010_formwizardhandler')]

    operations = [<AddField model_name='formentry', name='title',
field=<django.db.models.fields.CharField>>]
```

48.4.1.4.13 fobi.migrations.0012_auto_20161109_1550 module

```
class fobi.migrations.0012_auto_20161109_1550.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0011_formentry_title')]

    operations = [<AddField model_name='formwizardentry', name='title',
field=<django.db.models.fields.CharField>>, <AlterField model_name='formentry',
name='title', field=<django.db.models.fields.CharField>>]
```

48.4.1.4.14 fobi.migrations.0013_formwizardentry_show_all_navigation_buttons module

```
class fobi.migrations.0013_formwizardentry_show_all_navigation_buttons.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0012_auto_20161109_1550')]

    operations = [<AddField model_name='formwizardentry',
name='show_all_navigation_buttons', field=<django.db.models.fields.BooleanField>>]
```

48.4.1.4.15 fobi.migrations.0014_auto_20170516_1413 module

```
class fobi.migrations.0014_auto_20170516_1413.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0013_formwizardentry_show_all_navigation_buttons')]

    operations = [<AlterField model_name='formentry', name='slug',
field=<autoslug.fields.AutoSlugField>>, <AlterField model_name='formentry',
name='title', field=<django.db.models.fields.CharField>>, <AlterField
model_name='formwizardentry', name='slug', field=<autoslug.fields.AutoSlugField>>,
<AlterField model_name='formwizardentry', name='title',
field=<django.db.models.fields.CharField>>, <AlterField
model_name='formwizardentry', name='wizard_type',
field=<django.db.models.fields.CharField>>]
```

48.4.1.4.16 Module contents

48.4.1.5 fobi.reusable package

48.4.1.5.1 Subpackages

48.4.1.5.1.1 fobi.reusable.markdown_widget package

48.4.1.5.1.2 Submodules

48.4.1.5.1.3 fobi.reusable.markdown_widget.helpers module

```
fobi.reusable.markdown_widget.helpers.convert_to_markdown(content)
    Trans-compiles Markdown text to HTML.
```

Parameters `content` (*str*) – Markdown text.

Returns HTML encoded text.

Return type *str*

48.4.1.5.1.4 fobi.reusable.markdown_widget.widgets module

class `fobi.reusable.markdown_widget.widgets.MarkdownWidget`(*attrs=None*)

Bases: `django.forms.widgets.Textarea`

Markdown widget based on remarkable.

class `Media`

Bases: `object`

Media options.

```
js = ['markdown_widget/remarkable.min.js',
      'content_markdown/fobi.plugin.content_markdown.js']
```

property `media`

render(*name, value, attrs=None, **kwargs*)

Render the widget as an HTML string.

48.4.1.5.1.5 Module contents

48.4.1.5.2 Module contents

48.4.1.6 fobi.templatetags package

48.4.1.6.1 Submodules

48.4.1.6.2 fobi.templatetags.fobi_tags module

`fobi.templatetags.fobi_tags.get_fobi_form_handler_plugin_custom_actions`(*parser, token*)

Get the form handler plugin custom actions.

Note, that plugin shall be a instance of `fobi.models.FormHandlerEntry`.

Syntax

```
{% get_fobi_form_handler_plugin_custom_actions [plugin] [form_entry] as [con-
text_var_name] %}
```

Example

```
{% get_fobi_form_handler_plugin_custom_actions plugin form_entry as
form_handler_plugin_custom_actions %}
```

`fobi.templatetags.fobi_tags.get_fobi_form_wizard_handler_plugin_custom_actions`(*parser, token*)

Get the form wizard handler plugin custom actions.

Note, that plugin shall be a instance of `fobi.models.FormWizardHandlerEntry`.

Syntax


```
{% get_fobi_form_wizard_handler_plugin_custom_actions [plugin] [form_wizard_entry]
as [context_var_name] %}
```

Example

```
{% get_fobi_form_wizard_handler_plugin_custom_actions
    plugin form_wizard_entry as form_wizard_handler_plugin_custom_actions %}
```

`fobi.templatetags.fobi_tags.get_fobi_plugin(parser, token)`

Get the plugin.

Note, that entry shall be a instance of `fobi.models.FormElementEntry` or `fobi.models.FormHandlerEntry`.

Syntax `{% get_fobi_plugin entry as [context_var_name] %}`

Example `{% get_fobi_plugin entry as plugin %}`

```
{% get_fobi_plugin entry as plugin %} {{ plugin.render }}
```

`fobi.templatetags.fobi_tags.get_form_field_type(parser, token)`

Get form field type.

Syntax:

```
{% get_form_field_type [field] as [context_var_name] %}
```

Example:

```
{% get_form_field_type form.field as form_field_type %}
{% if form_field_type.is_checkbox %}
    ...
{% endif %}
```

`fobi.templatetags.fobi_tags.get_form_hidden_fields_errors(parser, token)`

Get form hidden fields errors.

Syntax `{% get_form_hidden_fields_errors [form] as [context_var_name] %}`

Example `{% get_form_hidden_fields_errors form as form_hidden_fields_errors %} {{
 form_hidden_fields_errors.as_ul }}`

`fobi.templatetags.fobi_tags.has_edit_form_entry_permissions(parser, token)`

Checks the permissions

Syntax `{% has_edit_form_entry_permissions as [var_name] %}`

Example `{% has_edit_form_entry_permissions %}`

or

```
{% has_edit_form_entry_permissions as has_permissions %}
```

`fobi.templatetags.fobi_tags.render_auth_link(context)`

Render auth link.

`fobi.templatetags.fobi_tags.render_fobi_forms_list(context, queryset, *args, **kwargs)`

Render the list of fobi forms.

Syntax `{% render_fobi_forms_list [queryset] [show_edit_link] [show_delete_link]
 [show_export_link] %}`

Example `{% render_fobi_forms_list queryset show_edit_link=True show_delete_link=False
 show_export_link=False %}`

48.4.1.6.3 fobi.template_tags.future_compat module

`fobi.template_tags.future_compat.firstof(parser, token)`

Output the first variable passed that is not False.

Output nothing if all the passed variables are False.

Sample usage:

```
{% firstof var1 var2 var3 as myvar %}
```

This is equivalent to:

```
{% if var1 %}
    {{ var1 }}
{% elif var2 %}
    {{ var2 }}
{% elif var3 %}
    {{ var3 }}
{% endif %}
```

but obviously much cleaner!

You can also use a literal string as a fallback value in case all passed variables are False:

```
{% firstof var1 var2 var3 "fallback value" %}
```

If you want to disable auto-escaping of variables you can use:

```
{% autoescape off %}
    {% firstof var1 var2 var3 "<strong>fallback value</strong>" %}
{% autoescape %}
```

Or if only some variables should be escaped, you can use:

```
{% firstof var1 var2|safe var3 "<strong>fallback value</strong>"|safe %}
```

48.4.1.6.4 Module contents

48.4.1.7 fobi.tests package

48.4.1.7.1 Submodules

48.4.1.7.2 fobi.tests.base module

class `fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest` (*methodName='runTest'*)

Bases: `django.contrib.staticfiles.testing.StaticLiveServerTestCase`

Browser tests django-fobi bulding forms functionality.

Backed up by selenium. This test is based on the bootstrap3 theme.

LIVE_SERVER_URL = `None`

cleans_up_after_itself = `True`

```

reset_sequences = True

classmethod setUpClass()
    Set up class.

take_screenshot(name='screenshot')

tearDown()
    Tear down.

classmethod tearDownClass()
    Tear down class.

```

48.4.1.7.3 fobi.tests.constants module

48.4.1.7.4 fobi.tests.core module

```

fobi.tests.core.is_app_setup_completed()
    Is fobi setup completed?

fobi.tests.core.mark_app_setup_as_completed()
    Mark fobi setup as completed.

fobi.tests.core.print_info(func)
    Prints some useful info.

fobi.tests.core.skip(func)
    Simply skips the test.

```

48.4.1.7.5 fobi.tests.data module

48.4.1.7.6 fobi.tests.helpers module

```

fobi.tests.helpers.create_form_with_entries(user=None, create_entries_if_form_exist=True, data={},
                                             is_public=False, name='Test form', slug='test-form')

```

Create test form with entries.

Fills the form with pre-defined plugins.

Parameters

- **user** (*django.contrib.auth.models.User*) –
- **create_entries_if_form_exist** (*bool*) – If set to True, entries are being created even if form already exists (a database record).
- **data** (*dict*) –
- **is_public** (*bool*) –
- **name** (*str*) –
- **slug** (*str*) –

Return *fobi.models.FormEntry* Instance of *fobi.models.FormEntry* with a number of form elements and handlers filled in.

```

fobi.tests.helpers.db_clean_up(clean_form=False, clean_elements=True, clean_handlers=True)
    Clean up the database.

```

Clean up the database by removing all form element and form handler entries.

`fobi.tests.helpers.get_or_create_admin_user()`

Create a user for testing the fobi.

TODO: At the moment an admin account is being tested. Automated tests with diverse accounts are to be implemented.

`fobi.tests.helpers.phantom_js_clean_up()`

Clean up Phantom JS.

Kills all phantomjs instances, disregard of their origin.

`fobi.tests.helpers.setup_app(collectstatic=False, fobi_sync_plugins=False)`

Set up fobi.

48.4.1.7.7 `fobi.tests.test_browser_build_dynamic_forms` module

`class fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuldDynamicFormsTest(methodName='runTest')`

Bases: `fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest`

Browser tests django-fobi bulding forms functionality.

Backed up by selenium. This test is based on the bootstrap3 theme.

`test_1001_open_dashboard(*args, **kwargs)`

Inner.

`test_2001_add_form(*args, **kwargs)`

Inner.

`test_2002_edit_form(*args, **kwargs)`

Inner.

`test_2003_delete_form(*args, **kwargs)`

Inner.

`test_2004_submit_form(*args, **kwargs)`

Inner.

`test_3001_add_form_elements(*args, **kwargs)`

Inner.

`test_3002_remove_form_elements(*args, **kwargs)`

Inner.

`test_3003_edit_form_elements(*args, **kwargs)`

Inner.

`test_4001_add_form_handlers(*args, **kwargs)`

Inner.

`test_4002_remove_form_handlers(*args, **kwargs)`

Inner.

`test_4003_edit_form_handlers(*args, **kwargs)`

Inner.

48.4.1.7.8 fobi.tests.test_core module

```
class fobi.tests.test_core.FobiCoreTest(methodName='runTest')
    Bases: django.test.testcases.TestCase
    Tests of django-fobi core functionality.

    setUp()
        Set up.

    test_01_get_registered_form_element_plugins(*args, **kwargs)
        Inner.

    test_02_get_registered_form_handler_plugins(*args, **kwargs)
        Inner.

    test_03_get_registered_form_callbacks(*args, **kwargs)
        Inner.

    test_04_get_registered_themes(*args, **kwargs)
        Inner.

    test_05_action_url(*args, **kwargs)
        Inner.

    test_06_form_entry_get_absolute_url(*args, **kwargs)
        Inner.

    test_07_form_wizard_entry_get_absolute_url(*args, **kwargs)
        Inner.

    test_08_form_entry_is_active(*args, **kwargs)
        Inner.
```

48.4.1.7.9 fobi.tests.test_drf_integration module

```
class fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTests(methodName='runTest')
    Bases: rest_framework.test.APITestCase
    DRF integration tests.

    setUp()
        Set up.

    classmethod setUpTestData()
        Load initial data for the TestCase

    tearDown()
        Set up.

    test_01_options_action_public_form()
        Test OPTIONS action call for public form.

    test_02_put_action_public_form()
        Test PUT action call for public form.

    test_03_fail_put_action_public_form()
        Test PUT action call fail test for public form.

    test_04_fail_options_action_non_public_form()
        Test OPTIONS action call fail test for non-public form.
```

```
test_05_fail_put_action()
    Test PUT action call fail test for non-public form.

test_06_options_action_non_public_form_auth_user()
    Test OPTIONS action call for authorised user for non-public form.

test_07_put_action_non_public_form_auth_user()
    Test PUT action call for authorised user for non-public form.

test_08_get_action_public_form()
    Test OPTIONS action call for public form.

test_09_get_action_non_public_form_auth_user()
    Test GET action call for authorised user for non-public form.

test_10_fail_get_action_non_public_form()
    Test GET action call fail test for non-public form.
```

48.4.1.7.10 fobi.tests.test_dynamic_forms module

```
class fobi.tests.test_dynamic_forms.FobiDynamicFormsTest(methodName='runTest')
    Bases: django.test.testcases.TestCase
    Tests of django-fob dynamic forms functionality.

    setUp()
        Set up.

    test_01_assemble_form_class_and_render_form(*args, **kwargs)
        Inner.
```

48.4.1.7.11 fobi.tests.test_feincms_integration module

```
class fobi.tests.test_feincms_integration.FeinCMSIntegrationTest(methodName='runTest')
    Bases: fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest
    FeinCMS integration tests.

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_fobi_form_widget_public_form()
        Test fobi form widget.
```

48.4.1.7.12 fobi.tests.test_form_importers_mailchimp module

```
class fobi.tests.test_form_importers_mailchimp.FormImportersMailchimpTest(methodName='runTest')
    Bases: django.test.testcases.TestCase
    Tests of form importers mailchimp functionality.

    setUp()
        Set up.

    test_01_test_mailchimp_importer(*args, **kwargs)
        Inner.
```

48.4.1.7.13 fobi.tests.test_sortable_dict module

```
class fobi.tests.test_sortable_dict.FobiDataStructuresTest(methodName='runTest')
    Bases: django.test.testcases.TestCase
    Tests of django-fobi data_structures module functionality.

    setUp()
        Set up.

    test_01_sortable_dict_move_before_key(*args, **kwargs)
        Inner.

    test_02_sortable_dict_move_after_key(*args, **kwargs)
        Inner.
```

48.4.1.7.14 Module contents

48.4.1.8 fobi.urls package

48.4.1.8.1 Submodules

48.4.1.8.2 fobi.urls.edit module

48.4.1.8.3 fobi.urls.view module

48.4.1.8.4 Module contents

48.4.1.9 fobi.wizard package

48.4.1.9.1 Subpackages

48.4.1.9.1.1 fobi.wizard.views package

48.4.1.9.1.2 Submodules

48.4.1.9.1.3 fobi.wizard.views.dynamic module

```
class fobi.wizard.views.dynamic.DynamicCookieWizardView(**kwargs)
    Bases: fobi.wizard.views.dynamic.DynamicWizardView
    A WizardView with pre-configured CookieStorage backend.

    storage_name = 'formtools.wizard.storage.cookie.CookieStorage'

class fobi.wizard.views.dynamic.DynamicNamedUrlCookieWizardView(**kwargs)
    Bases: fobi.wizard.views.dynamic.DynamicNamedUrlWizardView
    A NamedUrlFormWizard with pre-configured CookieStorageBackend.

    storage_name = 'formtools.wizard.storage.cookie.CookieStorage'

class fobi.wizard.views.dynamic.DynamicNamedUrlSessionWizardView(**kwargs)
    Bases: fobi.wizard.views.dynamic.DynamicNamedUrlWizardView
```

A NamedUrlWizardView with pre-configured SessionStorage backend.

```
storage_name = 'formtools.wizard.storage.session.SessionStorage'
```

```
class fobi.wizard.views.dynamic.DynamicNamedUrlWizardView(**kwargs)
```

Bases: [fobi.wizard.views.dynamic.DynamicWizardView](#)

A WizardView with URL named steps support.

```
done_step_name = None
```

```
get(*args, **kwargs)
```

GET request.

This renders the form or, if needed, does the http redirects.

```
get_context_data(form, **kwargs)
```

Get context data.

NamedUrlWizardView provides the `url_name` of this wizard in the context dict `wizard`.

```
classmethod get_initkwargs(*args, **kwargs)
```

Get init kwargs.

We require a `url_name` to reverse URLs later. Additionally users can pass a `done_step_name` to change the URL name of the “done” view.

```
get_step_url(step)
```

Get step URL.

```
post(*args, **kwargs)
```

POST request.

Do a redirect if user presses the prev. step button. The rest of this is super’d from WizardView.

```
render_done(form, **kwargs)
```

Render done.

When rendering the done view, we have to redirect first (if the URL name doesn’t fit).

```
render_goto_step(goto_step, **kwargs)
```

Render goto step.

This method gets called when the current step has to be changed. `goto_step` contains the requested step to go to.

```
render_next_step(form, **kwargs)
```

Render next step.

When using the NamedUrlWizardView, we have to redirect to update the browser’s URL to match the shown step.

```
render_revalidation_failure(failed_step, form, **kwargs)
```

Render revalidation failure.

When a step fails, we have to redirect the user to the first failing step.

```
url_name = None
```

```
class fobi.wizard.views.dynamic.DynamicSessionWizardView(**kwargs)
```

Bases: [fobi.wizard.views.dynamic.DynamicWizardView](#)

A WizardView with pre-configured SessionStorage backend.

```
storage_name = 'formtools.wizard.storage.session.SessionStorage'
```


class fobi.wizard.views.dynamic.DynamicWizardView(**kwargs)

Bases: django.views.generic.base.TemplateView

The WizardView is used to create multi-page forms.

Handles all the storage and validation stuff. The wizard is based on Django's generic class based views.

classmethod as_view(*args, **kwargs)

As view.

This method is used within urls.py to create unique wizardview instances for every request. We need to override this method because we add some kwargs which are needed to make the wizardview usable.

compute_form_list(form_list=None, *args, **kwargs)

Compute the forms list.

condition_dict = None

dispatch(request, *args, **kwargs)

Dispatch.

This method gets called by the routing engine. The first argument is *request* which contains a *HttpRequest* instance. The request is stored in *self.request* for later use. The storage instance is stored in *self.storage*.

After processing the request using the *dispatch* method, the response gets updated by the storage engine (for example add cookies).

done(form_list, **kwargs)

Done.

This method must be overridden by a subclass to process to form data after processing all steps.

get(request, *args, **kwargs)

GET requests.

This method handles GET requests.

If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to restart the process. The data of the wizard will be resetted before rendering the first step

get_all_cleaned_data()

Get all cleaned data.

Returns a merged dictionary of all step cleaned_data dictionaries. If a step contains a *FormSet*, the key will be prefixed with 'formset-' and contain a list of the formset cleaned_data dictionaries.

get_cleaned_data_for_step(step)

Get clean data for step.

Returns the cleaned data for a given *step*. Before returning the cleaned data, the stored values are revalidated through the form. If the data doesn't validate, None will be returned.

get_context_data(form, **kwargs)

Get context data.

Returns the template context for a step. You can overwrite this method to add more data for all or some steps. This method returns a dictionary containing the rendered form step. Available template context variables are:

- all extra data stored in the storage backend
- *wizard* - a dictionary representation of the wizard instance

Example:

```
class MyWizard(WizardView):
    def get_context_data(self, form, **kwargs):
        context = super(MyWizard, self).get_context_data(form=form,
                                                         **kwargs)

        if self.steps.current == 'my_step_name':
            context.update({'another_var': True})
        return context
```

get_form(*step=None, data=None, files=None*)

Get the form.

Constructs the form for a given *step*. If no *step* is defined, the current step will be determined automatically.

The form will be initialized using the *data* argument to prefill the new form. If needed, instance or queryset (for *ModelForm* or *ModelFormSet*) will be added too.

get_form_initial(*step*)

Get form initial

Returns a dictionary which will be passed to the form for *step* as *initial*. If no initial data was provided while initializing the form wizard, an empty dictionary will be returned.

get_form_instance(*step*)

Get form instance.

Returns an object which will be passed to the form for *step* as *instance*. If no instance object was provided while initializing the form wizard, None will be returned.

get_form_kwargs(*step=None*)

Get form kwargs.

Returns the keyword arguments for instantiating the form (or formset) on the given step.

get_form_list()

Get form list.

This method returns a *form_list* based on the initial form list but checks if there is a condition method/value in the *condition_list*. If an entry exists in the condition list, it will call/read the value and respect the result. (True means add the form, False means ignore the form)

The *form_list* is always generated on the fly because condition methods could use data from other (maybe previous forms).

get_form_prefix(*step=None, form=None*)

Get form prefix.

Returns the prefix which will be used when calling the actual form for the given step. *step* contains the step-name, *form* the form which will be called with the returned prefix.

If no step is given, the *form_prefix* will determine the current step automatically.

get_form_step_data(*form*)

Get form step data.

Is used to return the raw form data. You may use this method to manipulate the data.

get_form_step_files(*form*)

Get form step files.

Is used to return the raw form files. You may use this method to manipulate the data.

get_initial_wizard_data(**args, **kwargs*)

This should be implemented in your subclass.

You are supposed to return a dict with the dynamic properties, such as *form_list* or *template_name*.

classmethod `get_initkwargs`(*form_list=None, initial_dict=None, instance_dict=None, condition_dict=None, *args, **kwargs*)

Create a dict with all needed parameters.

For the form wizard instances.

- *form_list* - is a list of forms. The list entries can be single form classes or tuples of (*step_name, form_class*). If you pass a list of forms, the wizardview will convert the class list to (*zero_based_counter, form_class*). This is needed to access the form for a specific step.
- *initial_dict* - contains a dictionary of initial data dictionaries. The key should be equal to the *step_name* in the *form_list* (or the str of the zero based counter - if no *step_names* added in the *form_list*)
- *instance_dict* - contains a dictionary whose values are model instances if the step is based on a `ModelForm` and querysets if the step is based on a `ModelFormSet`. The key should be equal to the *step_name* in the *form_list*. Same rules as for *initial_dict* apply.
- *condition_dict* - contains a dictionary of boolean values or callables. If the value of for a specific *step_name* is callable it will be called with the wizardview instance as the only argument. If the return value is true, the step's form will be used.

get_next_step(*step=None*)

Get next step.

Returns the next step after the given *step*. If no more steps are available, `None` will be returned. If the *step* argument is `None`, the current step will be determined automatically.

get_prefix(*request, *args, **kwargs*)

Get prefix.

get_prev_step(*step=None*)

Get previous step.

Returns the previous step before the given *step*. If there are no steps available, `None` will be returned. If the *step* argument is `None`, the current step will be determined automatically.

get_step_index(*step=None*)

Get step index.

Returns the index for the given *step* name. If no step is given, the current step will be used to get the index.

initial_dict = `None`

instance_dict = `None`

post(**args, **kwargs*)

POST requests.

This method handles POST requests.

The wizard will render either the current step (if form validation wasn't successful), the next step (if the current step was stored successful) or the done view (if no more steps are available)

process_step(*form*)

Process the step.

This method is used to post-process the form data. By default, it returns the raw *form.data* dictionary.

process_step_files(*form*)

Process step files.

This method is used to post-process the form files. By default, it returns the raw *form.files* dictionary.

render(*form=None*, ***kwargs*)

Render.

Returns a `HttpResponse` containing all needed context data.

render_done(*form*, ***kwargs*)

Render done.

This method gets called when all forms passed. The method should also re-validate all steps to prevent manipulation. If any form fails to validate, *render_revalidation_failure* should get called. If everything is fine call *done*.

render_goto_step(*goto_step*, ***kwargs*)

Render goto step.

This method gets called when the current step has to be changed. *goto_step* contains the requested step to go to.

render_next_step(*form*, ***kwargs*)

Render next step.

This method gets called when the next step/form should be rendered. *form* contains the last/current form.

render_revalidation_failure(*step*, *form*, ***kwargs*)

Render revalidation failure.

Gets called when a form doesn't validate when rendering the done view. By default, it changes the current step to failing forms step and renders the form.

storage_name = `None`

template_name = `'formtools/wizard/wizard_form.html'`

48.4.1.9.1.4 fobi.wizard.views.views module

class `fobi.wizard.views.views.CookieWizardView`(***kwargs*)

Bases: `formtools.wizard.views.CookieWizardView`, `fobi.wizard.views.views.PatchGetMixin`

A `WizardView` with pre-configured `CookieStorage` backend.

get(*request*, **args*, ***kwargs*)

GET requests.

This method handles GET requests.

If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to restart the process. The data of the wizard will be resetted before rendering the first step.

class `fobi.wizard.views.views.SessionWizardView`(***kwargs*)

Bases: `formtools.wizard.views.SessionWizardView`, `fobi.wizard.views.views.PatchGetMixin`

A `WizardView` with pre-configured `SessionStorage` backend.

get(*request*, **args*, ***kwargs*)

GET requests.

This method handles GET requests.

If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to restart the process. The data of the wizard will be resetted before rendering the first step.

```
class fobi.wizard.views.views.WizardView(**kwargs)
    Bases: formtools.wizard.views.WizardView, fobi.wizard.views.views.PatchGetMixin

    Patched version of the original WizardView.

    get(request, *args, **kwargs)
        GET requests.

        This method handles GET requests.

        If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to
        restart the process. The data of the wizard will be resetted before rendering the first step.
```

48.4.1.9.1.5 Module contents

48.4.1.9.2 Module contents

48.4.2 Submodules

48.4.3 fobi.admin module

```
class fobi.admin.BasePluginModelAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    Base plugin admin.

    class Meta
        Bases: object

        Meta.

        app_label = 'Fobi'

    bulk_change_plugins(request)
        Bulk change plugins.

        This is where the data is actually processed.

    fieldsets = ((None, {'fields': ('plugin_uid', 'users', 'groups')})),)
    filter_horizontal = ('users', 'groups')

    get_queryset(request)
        Tweak queryset.

    has_add_permission(request)
        Has add permissions.

        We don't want to allow to add form elements/handlers manually. It should happen using the management
        command fobi_sync_plugins instead.

    list_display = ('plugin_uid_admin', 'users_list', 'groups_list')
    property media
    readonly_fields = ('plugin_uid', 'plugin_uid_admin')

class fobi.admin.FormElementAdmin(model, admin_site)
    Bases: fobi.admin.BasePluginModelAdmin

    FormElement admin.

    actions = [<function bulk_change_form_element_plugins>]
```

```

    get_urls()
        Get URLs.

    property media

class fobi.admin.FormElementEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    FormElementEntry admin.

    class Meta
        Bases: object

        Meta.

        app_label = 'Fobi'

    fieldsets = (('Plugin', {'fields': ('plugin_uid', 'plugin_data')}), ('Form',
{'fields': ('form_entry', 'form_fieldset_entry', 'position')}))

    get_queryset(request)
        Get queryset.

    list_display = ('plugin_uid', 'plugin_uid_code', 'plugin_data', 'position',
'form_entry')

    list_editable = ('position',)

    list_filter = ('form_entry', 'plugin_uid')

    property media

    readonly_fields = ('plugin_uid_code',)

class fobi.admin.FormElementEntryInlineAdmin(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    FormElementEntry inline admin.

    extra = 0

    fields = ('form_entry', 'plugin_uid', 'plugin_data', 'position')

    form
        alias of fobi.forms.FormElementEntryForm

    property media

    model
        alias of fobi.models.FormElementEntry

class fobi.admin.FormEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    FormEntry admin.

    class Meta
        Bases: object

        Meta.

        app_label = 'Fobi'

```

```

fieldsets = (('Form', {'fields': ('name', 'is_public', 'is_cloneable',
'active_date_from', 'active_date_to', 'inactive_page_title',
'inactive_page_message')}), ('Custom', {'classes': ('collapse',), 'fields':
('success_page_title', 'success_page_message', 'action')}), ('User', {'classes':
('collapse',), 'fields': ('user',)}), ('Additional', {'classes': ('collapse',),
'fields': ('slug',)}))

inlines = [<class 'fobi.admin.FormElementEntryInlineAdmin'>, <class
'fobi.admin.FormHandlerEntryInlineAdmin'>]

list_display = ('name', 'slug', 'user', 'is_public', 'is_active', 'created',
'updated', 'is_cloneable')

list_editable = ('is_public', 'is_cloneable')

list_filter = ('is_public', 'is_cloneable')

property media

radio_fields = {'user': 2}

readonly_fields = ('slug',)

class fobi.admin.FormFieldsetEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    FormFieldsetEntry admin.

    class Meta
        Bases: object
        Meta.
        app_label = 'Fobi'

    fieldsets = ((None, {'fields': ('form_entry', 'name', 'is_repeatable')}),)
    list_display = ('form_entry', 'name', 'is_repeatable')
    list_editable = ('is_repeatable',)
    list_filter = ('is_repeatable',)
    property media

class fobi.admin.FormHandlerAdmin(model, admin_site)
    Bases: fobi.admin.BasePluginModelAdmin
    FormHandler admin.

    actions = [<function bulk_change_form_handler_plugins>]
    get_urls()
        Get URLs.
    property media

class fobi.admin.FormHandlerEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    FormHandlerEntry admin.

    class Meta
        Bases: object
        Meta.

```

```

        app_label = 'Form handler entry'

    fieldsets = (('Plugin', {'fields': ('plugin_uid', 'plugin_data')}), ('Form',
{'fields': ('form_entry',)}))

    get_queryset(request)
        Tweak the queryset.

    list_display = ('plugin_uid', 'plugin_uid_code', 'plugin_data', 'form_entry')

    list_filter = ('form_entry', 'plugin_uid')

    property media

    readonly_fields = ('plugin_uid_code',)

class fobi.admin.FormHandlerEntryInlineAdmin(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    FormHandlerEntry inline admin.

    extra = 0

    fields = ('form_entry', 'plugin_uid', 'plugin_data')

    form
        alias of fobi.forms.FormHandlerEntryForm

    property media

    model
        alias of fobi.models.FormHandlerEntry

class fobi.admin.FormWizardEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    FormWizardEntry admin.

    class Meta
        Bases: object

        Meta.

        app_label = 'Fobi'

    fieldsets = (('Form', {'fields': ('name', 'is_public', 'is_cloneable')}),
('Custom', {'classes': ('collapse',), 'fields': ('success_page_title',
'success_page_message')}), ('User', {'classes': ('collapse',), 'fields':
('user',)}), ('Additional', {'classes': ('collapse',), 'fields': ('slug',)}))

    inlines = [<class 'fobi.admin.FormWizardFormEntryInlineAdmin'>, <class
'fobi.admin.FormWizardHandlerEntryInlineAdmin'>]

    list_display = ('name', 'slug', 'user', 'is_public', 'created', 'updated',
'is_cloneable')

    list_editable = ('is_public', 'is_cloneable')

    list_filter = ('is_public', 'is_cloneable')

    property media

    radio_fields = {'user': 2}

    readonly_fields = ('slug',)

```



```

class fobi.admin.FormWizardFormEntryInlineAdmin(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    FormWizardFormEntry inline admin.

    extra = 0

    fields = ('form_entry', 'position')

    property media

    model
        alias of fobi.models.FormWizardFormEntry

class fobi.admin.FormWizardHandlerAdmin(model, admin_site)
    Bases: fobi.admin.BasePluginModelAdmin

    FormHandler admin.

    actions = [<function bulk_change_form_wizard_handler_plugins>]

    get_urls()
        Get URLs.

    property media

class fobi.admin.FormWizardHandlerEntryInlineAdmin(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    FormWizardHandlerEntry inline admin.

    extra = 0

    fields = ('plugin_uid', 'plugin_data')

    form
        alias of fobi.forms.FormWizardHandlerEntryForm

    property media

    model
        alias of fobi.models.FormWizardHandlerEntry

fobi.admin.base_bulk_change_plugins(PluginForm, named_url, modeladmin, request, queryset)
    Bulk change of plugins action additional view.

fobi.admin.bulk_change_form_element_plugins(modeladmin, request, queryset)
    Bulk change FormElement plugins.

fobi.admin.bulk_change_form_handler_plugins(modeladmin, request, queryset)
    Bulk change FormHandler plugins.

fobi.admin.bulk_change_form_wizard_handler_plugins(modeladmin, request, queryset)
    Bulk change FormWizardHandler plugins.

```

48.4.4 fobi.app module

`fobi.app.app_config(path, config_app_path='apps.Config')`
App config.

Parameters

- **path** (*str*) – Absolute app path (from project root).
- **config_app_path** (*str*) – Relative config path (from app root)

Return str

`fobi.app.app_name(path, reduce_depth_by=1)`
Return another path by reducing the depth by one.

Parameters

- **path** (*str*) – Absolute app path (from project root).
- **reduce_depth_by** (*int*) –

Return str

48.4.5 fobi.apps module

`class fobi.apps.Config(app_name, app_module)`
Bases: `django.apps.config AppConfig`
Config.
label = 'fobi'
name = 'fobi'

48.4.6 fobi.base module

Base module. All *uids* are supposed to be pythonic function names (see PEP <http://www.python.org/dev/peps/pep-0008/#function-names>).

`class fobi.base.BaseDataStorage`
Bases: `object`
Base storage data.

`class fobi.base.BaseFormFieldPluginForm`
Bases: `fobi.base.BasePluginForm`
Base form for form field plugins.
help_text = <django.forms.fields.CharField object>
label = <django.forms.fields.CharField object>
name = <django.forms.fields.CharField object>
plugin_data_fields = [('name', ''), ('label', ''), ('help_text', ''), ('required', False)]
required = <django.forms.fields.BooleanField object>
validate_plugin_data(*form_element_entries*, *request=None*)
Validate plugin data.

Parameters

- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry`.
- **request** (*django.http.HttpRequest*) –

Return bool

class `fobi.base.BasePlugin`(*user=None*)

Bases: `object`

Base plugin.

Base form field from which every form field should inherit.

Properties

- **uid** (*string*): **Plugin uid (obligatory).** Example value: ‘dummy’, ‘wysiwyg’, ‘news’.
- **name** (*string*): **Plugin name (obligatory).** Example value: ‘Dummy plugin’, ‘WYSIWYG’, ‘Latest news’.
- **description** (*string*): **Plugin decription (optional).** Example value: ‘Dummy plugin used just for testing’.
- **help_text** (*string*): **Plugin help text (optional).** This text would be shown in `fobi.views.add_form_plugin_entry` and `fobi.views.edit_form_plugin_entry` views.
- **form**: **Plugin form (optional).** A subclass of `django.forms.Form`. Should be given in case plugin is configurable.
- **add_form_template** (*str*) (*optional*): **Add form template (optional).** If given, overrides the `fobi.views.add_form_handler_entry` default template.
- **edit_form_template** (*string*): **Edit form template (optional).** If given, overrides the `fobi.views.edit_form_handler_entry` default template.
- **html_classes** (*list*): List of extra HTML classes for the plugin.
- **group** (*string*): **Plugin are grouped under the specified group.** Override in your plugin if necessary.

add_form_template = `None`

clone_plugin_data(*entry*)

Clone plugin data.

Used when copying entries. If any objects or files are created by plugin, they should be cloned.

Parameters `fobi.models.AbstractPluginEntry` – Instance of `fobi.models.AbstractPluginEntry`.

Return string JSON dumped string of the cloned plugin data. The returned value would be inserted as is into the `fobi.models.AbstractPluginEntry.plugin_data` field.

delete_plugin_data()

Delete plugin data (internal method).

Used in `fobi.views.delete_form_entry` and `fobi.views.delete_form_handler_entry`. Fired automatically, when `fobi.models.FormEntry` object is about to be deleted. Make use of it if your plugin creates database records or files that are not monitored externally but by fobi only.

description = `None`

`edit_form_template = None`

`form = None`

`get_cloned_plugin_data(update={})`

Get cloned plugin data.

Get the cloned plugin data and returns it in a JSON dumped format.

Parameters `update (dict)` –

Return string JSON dumped string of the cloned plugin data.

Example

In the `get_cloned_plugin_data` method of your plugin, do as follows:

```
>>> def clone_plugin_data(self, dashboard_entry):
>>>     cloned_image = clone_file(self.data.image, relative_path=True)
>>>     return self.get_cloned_plugin_data(
>>>         update={'image': cloned_image}
>>>     )
```

`get_form()`

Get the plugin form class.

Override this method in your subclassed `fobi.base.BasePlugin` class when you need your plugin setup to vary depending on the placeholder, workspace, user or request given. By default returns the value of the `form` attribute defined in your plugin.

Return `django.forms.Form|django.forms.ModelForm` Subclass of `django.forms.Form` or `django.forms.ModelForm`.

`get_initialised_create_form(data=None, files=None, initial_data=None)`

Get initialized create form.

Used `fobi.views.add_form_element_entry` and `fobi.views.add_form_handler_entry` view to gets initialised form for object to be created.

`get_initialised_create_form_or_404(data=None, files=None)`

Get initialized create form or page 404.

Same as `get_initialised_create_form` but raises `django.http.Http404` on errors.

`get_initialised_edit_form(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=':', empty_permitted=False, instance=None)`

Get initialized edit form.

Used in `fobi.views.edit_form_element_entry` and `fobi.views.edit_form_handler_entry` views.

`get_initialised_edit_form_or_404(data=None, files=None, auto_id='id_%s', prefix=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=':', empty_permitted=False)`

Get initialized edit form or page 404.

Same as `get_initialised_edit_form` but raises `django.http.Http404` on errors.

`get_instance()`

Get instance.

get_plugin_form_data()

Get plugin form data.

Fed as `initial` argument to the plugin form when initialising the instance for adding or editing the plugin. Override in your plugin class if you need customisations.

get_updated_plugin_data(*update={}*)

Get updated plugin data.

Returns it in a JSON dumped format.

Parameters `update` (*dict*) –

Return string JSON dumped string of the cloned plugin data.

get_widget(*request=None, as_instance=False*)

Get the plugin widget.

Parameters

- **request** (*django.http.HttpRequest*) –
- **as_instance** (*bool*) –

Return mixed Subclass of *fobi.base.BasePluginWidget* or instance of subclassed *fobi.base.BasePluginWidget* object.

group = 'General'

help_text = None

property html_class

HTML class.

A massive work on positioning the plugin and having it to be displayed in a given width is done here. We should be getting the plugin widget for the plugin given and based on its' properties (static!) as well as on plugin position (which we have from model), we can show the plugin with the exact class.

html_classes = []

property html_id

HTML id.

load_plugin_data(*plugin_data*)

Load plugin data.

Load the plugin data saved in *fobi.models.FormElementEntry* or *fobi.models.FormHandlerEntry*. Plugin data is saved in JSON string.

Parameters `plugin_data` (*string*) – JSON string with plugin data.

media_css = []

media_js = []

name = None

plugin_data_repr()

Plugin data repr.

Human readable representation of plugin data. A very basic way would be just:

```
>>> return self.data.__dict__
```

Return string

post_processor()

Post-processor (self).

Redefine in your subclassed plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (django.http.HttpRequest) is available (self.request).

pre_processor()

Pre-processor (callback).

Redefine in your subclassed plugin when necessary.

Pre process plugin data (before rendering). This method is being called before the data has been loaded into the plugin.

Note, that request (django.http.HttpRequest) is available (self.request).

process(plugin_data=None, fetch_related_data=False)

Process.

Init plugin with data.

process_plugin_data(fetch_related_data=False)

Processes plugin data.

render(request=None)

Renders the plugin HTML.

Parameters **request** (django.http.HttpRequest) –

Return string

storage = None

uid = None

update_plugin_data(entry)

Update plugin data.

Used in fobi.management.commands.fobi_update_plugin_data.

Some plugins would contain data fetched from various sources (models, remote data). Since form entries are by definition loaded extremely much, you are advised to store as much data as possible in `plugin_data` field of `fobi.models.FormElementEntry` or `fobi.models.FormHandlerEntry`. Some externally fetched data becomes invalid after some time and needs updating. For that purpose, in case if your plugin needs that, redefine this method in your plugin. If you need your data to be periodically updated, add a cron-job which would run `fobi_update_plugin_data` management command (see `fobi.management.commands.fobi_update_plugin_data` module).

Parameters or fobi.models.FormHandlerEntry (fobi.models.FormElementEntry) –

Instance of `fobi.models.FormeHandlerEntry`.

Return dict Should return a dictionary containing data of fields to be updated.

widget = None

class fobi.base.BasePluginForm

Bases: object

Not a form actually; defined for magic only.

Property iterable plugin_data_fields Fields to get when calling the `get_plugin_data` method. These field will be JSON serialized. All other fields, even if they are part of the form, won't be. Make sure all fields are serializable. If some of them aren't, override the `save_plugin_data` method and make them serializable there. See `fobi.contrib.plugins.form_elements.fields.select.forms` as a good example.

Example

```
>>> plugin_data_fields = (
>>>     ('name', ''),
>>>     ('active': False)
>>> )
```

get_plugin_data(*request=None, json_format=True*)

Get plugin data.

Data that would be saved in the `plugin_data` field of the `fobi.models.FormElementEntry` or `fobi.models.FormHandlerEntry` subclassed model.`

Parameters *request* (`django.http.HttpRequest`) –

plugin_data_fields = `None`

save_plugin_data(*request=None*)

Save plugin data.

Dummy, but necessary.

validate_plugin_data(*form_element_entries, request=None*)

Validate plugin data.

Parameters

- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry`.
- **request** (`django.http.HttpRequest`) –

Return bool

class `fobi.base.BaseRegistry`

Bases: `object`

Base registry.

Registry of plugins. It's essential, that class registered has the `uid` property.

If `fail_on_missing_plugin` is set to `True`, an appropriate exception (`plugin_not_found_exception_cls`) is raised in cases if plugin could't be found in the registry.

Property mixed type

Property bool fail_on_missing_plugin

Property `fobi.exceptions.DoesNotExist` `plugin_not_found_exception_cls`

Property str plugin_not_found_error_message

fail_on_missing_plugin = `False`

get(*uid, default=None*)

Get the given entry from the registry.

Parameters

- **uid** (*string*) –

- **default** (*mixed*) –

:return mixed.

items()
Shortcut to self._registry.items().

plugin_not_found_error_message = "Can't find plugin with uid `{0}` in `{1}` registry."

plugin_not_found_exception_cls
alias of `fobi.exceptions.DoesNotExist`

register(*cls*, *force=False*)
Registers the plugin in the registry.

Parameters

- **cls** (*mixed*) –
- **force** (*bool*) –

property registry
Shortcut to self._registry.

type = None

unregister(*cls*)
Un-register.

class `fobi.base.ClassProperty`(*fget=None*, *fset=None*, *fdel=None*, *doc=None*)
Bases: `property`
ClassProperty.

class `fobi.base.FormCallback`
Bases: `fobi.base.BaseFormCallback`
Form callback.

callback(*form_entry*, *request*, *form*)
Callback.
Custom callback code should be implemented here.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

class `fobi.base.FormCallbackRegistry`
Bases: `object`
Registry of callbacks.
Holds callbacks for stages listed in the `fobi.constants.CALLBACK_STAGES`.

get_callbacks(*stage=None*)
Get callbacks for the stage given.

Parameters **stage** (*string*) –

Return list

register(*cls*)

Registers the plugin in the registry.

Parameters *cls* (*mixed*) –

uidfy(*cls*)

Makes a UID string from the class given.

Parameters *cls* (*mixed*) –

Return string

class `fobi.base.FormElementPlugin`(*user=None*)

Bases: `fobi.base.BasePlugin`

Base form element plugin.

Property `fobi.base.FormElementPluginDataStorage` **storage**

Property `bool` **has_value** If set to False, ignored (removed) from the POST when processing the form.

get_custom_field_instances(*integrate_with*, *request=None*, *form_entry=None*,
form_element_entries=None, *has_value=None*, ***kwargs*)

Get custom field instances.

Parameters

- **integrate_with** (*str*) –
- **request** (`django.http.HttpRequest`) –
- **form_entry** –
- **form_element_entries** –
- **has_value** (*bool*) – If not None, used for filtering out.

Return list

get_form_field_instances(*request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get the instances of form fields, that plugin contains.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry** (`fobi.models.FormEntry`) –
- **form_element_entries** (`django.db.models.QuerySet`) – Queryset of `fobi.models.FormElementEntry` instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_origin_kwargs_update_func_results(*kwargs_update_func*, *form_element_entry*, *origin*, *extra={}*,
widget_cls=None)

Get origin kwargs update func results.

If *kwargs_update_func* is given, is callable and returns results without failures, return the result. Otherwise - return None.

get_origin_return_func_results(*return_func*, *form_element_entry*, *origin*)

Get origin return func results.

If *return_func* is given, is callable and returns results without failures, return the result. Otherwise - return None.

has_value = False

is_hidden = False

storage

alias of *fobi.base.FormElementPluginDataStorage*

submit_plugin_form_data(*form_entry*, *request*, *form*, *form_element_entries*=None, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –
- **form_element_entries** (*iterable*) –

class *fobi.base.FormElementPluginDataStorage*

Bases: *fobi.base.BaseDataStorage*

Storage for *FormElementPlugin* data.

class *fobi.base.FormElementPluginRegistry*

Bases: *fobi.base.BaseRegistry*

Form element plugins registry.

fail_on_missing_plugin = True

plugin_not_found_exception_cls

alias of *fobi.exceptions.FormElementPluginDoesNotExist*

type = (<class 'fobi.base.FormElementPlugin'>, <class 'fobi.base.FormFieldPlugin'>)

class *fobi.base.FormElementPluginWidget(plugin)*

Bases: *fobi.base.BasePluginWidget*

Form element plugin widget.

storage

alias of *fobi.base.FormElementPluginWidgetDataStorage*

class *fobi.base.FormElementPluginWidgetRegistry*

Bases: *fobi.base.BasePluginWidgetRegistry*

Registry of form element plugins.

type

alias of *fobi.base.FormElementPluginWidget*

class *fobi.base.FormFieldPlugin(user=None)*

Bases: *fobi.base.FormElementPlugin*

Form field plugin.

has_value = True

class fobi.base.**FormHandlerPlugin**(*user=None*)

Bases: [fobi.base.BasePlugin](#)

Form handler plugin.

Property fobi.base.**FormHandlerPlugin**.**data_storage** **storage**

Property bool **allow_multiple** If set to True, plugin can be used multiple times within (per form). Otherwise - just once.

allow_multiple = True

custom_actions(*form_entry, request=None*)

Custom actions.

Override this method in your form handler if you want to specify custom actions. Note, that expected return value of this method is an iterable with a triple, where the first item is the URL of the action and the second item is the action title and the third item is the icon class of the action.

Example

```
>>> return (
>>>     ('/add-to-favorites/',
>>>      'Add to favourites',
>>>      'glyphicon glyphicon-favourties'),
>>> )
```

get_custom_actions(*form_entry, request=None*)

Internal method to for obtaining the **get_custom_actions**.

run(*form_entry, request, form, form_element_entries=None*)

Run.

Custom code should be implemented here.

Parameters

- **form_entry** ([fobi.models.FormEntry](#)) – Instance of [fobi.models.FormEntry](#).
- **request** ([django.http.HttpRequest](#)) –
- **form** ([django.forms.Form](#)) –
- **form_element_entries** (*iterable*) – Iterable of [fobi.models.FormElementEntry](#) objects.

Return mixed May be a tuple (bool, mixed) or None

run_integration_handler(*integrate_with, form_entry, request, form_element_entries=None, **kwargs*)

Run integration handler.

storage

alias of [fobi.base.FormHandlerPluginDataStorage](#)

class fobi.base.**FormHandlerPluginDataStorage**

Bases: [fobi.base.BaseDataStorage](#)

Storage for *FormHandlerPlugin* data.

class fobi.base.**FormHandlerPluginRegistry**

Bases: [fobi.base.BaseRegistry](#)

Form handler plugins registry.

fail_on_missing_plugin = True

plugin_not_found_exception_cls
alias of `fobi.exceptions.FormHandlerPluginDoesNotExist`

type
alias of `fobi.base.FormHandlerPlugin`

class `fobi.base.FormHandlerPluginWidget(plugin)`

Bases: `fobi.base.BasePluginWidget`

Form handler plugin widget.

storage
alias of `fobi.base.FormHandlerPluginWidgetDataStorage`

class `fobi.base.FormHandlerPluginWidgetRegistry`

Bases: `fobi.base.BasePluginWidgetRegistry`

Registry of form handler plugins.

type
alias of `fobi.base.FormHandlerPluginWidget`

class `fobi.base.FormWizardHandlerPlugin(user=None)`

Bases: `fobi.base.BasePlugin`

Form wizard handler plugin.

Property `fobi.base.FormWizardHandlerPluginDataStorage storage`

Property `bool allow_multiple` If set to True, plugin can be used multiple times within (per form).
Otherwise - just once.

DONE

allow_multiple = True

custom_actions(`form_wizard_entry, request=None`)

Custom actions.

Override this method in your form handler if you want to specify custom actions. Note, that expected return value of this method is an iterable with a triple, where the first item is the URL of the action and the second item is the action title and the third item is the icon class of the action.

Example

```
>>> return (
>>>     ('/add-to-favorites/',
>>>      'Add to favourites',
>>>      'glyphicon glyphicon-favourties'),
>>> )
```

get_custom_actions(`form_wizard_entry, request=None`)

Internal method to for obtaining the `get_custom_actions`.

run(`form_wizard_entry, request, form_list, form_wizard, form_element_entries=None`)

Run.

Custom code should be implemented here.

Parameters

- **form_wizard_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –

- **form** (*django.forms.Form*) –
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

Return mixed May be a tuple (bool, mixed) or None

storage

alias of *fobi.base.FormWizardHandlerPluginDataStorage*

class *fobi.base.FormWizardHandlerPluginDataStorage*

Bases: *fobi.base.BaseDataStorage*

Storage for *FormWizardHandlerPlugin* handler data.

class *fobi.base.FormWizardHandlerPluginRegistry*

Bases: *fobi.base.BaseRegistry*

Form wizard handler plugins registry.

fail_on_missing_plugin = True

plugin_not_found_exception_cls

alias of *fobi.exceptions.FormWizardHandlerPluginDoesNotExist*

type

alias of *fobi.base.FormWizardHandlerPlugin*

class *fobi.base.FormWizardHandlerPluginWidget(plugin)*

Bases: *fobi.base.BasePluginWidget*

Form wizard handler plugin widget.

storage

alias of *fobi.base.FormWizardHandlerPluginWidgetDataStorage*

class *fobi.base.FormWizardHandlerPluginWidgetRegistry*

Bases: *fobi.base.BasePluginWidgetRegistry*

Registry of form wizard handler plugins.

type

alias of *fobi.base.FormWizardHandlerPluginWidget*

class *fobi.base.IntegrationFormCallback*

Bases: object

Integration form callback.

callback(*form_entry, request, **kwargs*)

Callback.

Custom callback code should be implemented here.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

integrate_with = None

class *fobi.base.IntegrationFormCallbackRegistry*

Bases: object

Registry of callbacks for integration plugins.

Holds callbacks for stages listed in the `fobi.constants.CALLBACK_STAGES`.

get_callbacks(*integrate_with*, *stage=None*)

Get callbacks for the stage given.

Parameters

- **integrate_with** (*str*) –
- **stage** (*string*) –

Return list

register(*cls*)

Registers the plugin in the registry.

Parameters **cls** (*mixed*) –

property registry

uidfy(*cls*)

Makes a UID string from the class given.

Parameters **cls** (*mixed*) –

Return string

class `fobi.base.IntegrationFormElementPlugin`(*user=None*)

Bases: `fobi.base.BasePlugin`

Base custom field instance plugin for integration.

has_value = `False`

is_hidden = `False`

storage

alias of `fobi.base.IntegrationFormElementPluginDataStorage`

class `fobi.base.IntegrationFormElementPluginDataStorage`

Bases: `fobi.base.BaseDataStorage`

Storage for `IntegrationFormElementPlugin`.

class `fobi.base.IntegrationFormElementPluginProcessor`(*args, **kwargs)

Bases: `object`

Custom form field instance processor.

Supposed to have implemented a single method called `process_custom_form_field_instance`.

process_custom_form_field_instance(*form_element_entry*, *form_entry*, *request*, *form_element_plugin*)

You should implement this method in your implementation.

class `fobi.base.IntegrationFormElementPluginRegistry`

Bases: `fobi.base.BaseIntegrationPluginRegistry`

Integration form element plugin registry.

fail_on_missing_plugin = `True`

plugin_not_found_exception_cls

alias of `fobi.exceptions.IntegrationFormElementPluginDoesNotExist`

type = (`<class 'fobi.base.IntegrationFormElementPlugin'>`,)

```
class fobi.base.IntegrationFormFieldPlugin(user=None)
    Bases: fobi.base.IntegrationFormElementPlugin

    Integration form field plugin for custom field instances.

    has_value = True

class fobi.base.IntegrationFormHandlerPlugin(user=None)
    Bases: fobi.base.BasePlugin

    Base integration form handler plugin for integration.

    has_value = True
    is_hidden = False
    storage
        alias of fobi.base.IntegrationFormHandlerPluginDataStorage

class fobi.base.IntegrationFormHandlerPluginDataStorage
    Bases: fobi.base.BaseDataStorage

    Storage for IntegrationFormHandlerPlugin.

class fobi.base.IntegrationFormHandlerPluginRegistry
    Bases: fobi.base.BaseIntegrationPluginRegistry

    Integration form handler plugin registry.

    fail_on_missing_plugin = True
    plugin_not_found_exception_cls
        alias of fobi.exceptions.IntegrationFormHandlerPluginDoesNotExist
    type = (<class 'fobi.base.IntegrationFormHandlerPlugin'>,)

fobi.base.assemble_form_field_widget_class(base_class, plugin)
    Assemble form field widget class.

    Finish this or remove.

    #TODO

fobi.base.classproperty
    alias of fobi.base.ClassProperty

fobi.base.collect_plugin_media(form_element_entries, request=None)
    Collect the plugin media for form element entries given.

    Parameters
        • form_element_entries (iterable) – Iterable of fobi.models.FormElementEntry instances.
        • request (django.http.HttpRequest) –

    Return dict
        Returns a dict containing the 'js' and 'css' keys. Correspondent values of those keys are lists containing paths to the CSS and JS media files.

fobi.base.ensure_autodiscover()
    Ensure that plugins are auto-discovered.

    The form callbacks registry is intentionally left out, since they will be auto-discovered in any case if other modules are discovered.

fobi.base.fire_form_callbacks(form_entry, request, form, stage=None)
    Fire form callbacks.
```

Parameters

- **form_entry** (`fobi.models.FormEntry`) –
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **stage** (`string`) –

Return `django.forms.Form` form

`fobi.base.get_form_element_plugin_widget(plugin_uid, request=None, as_instance=False, theme=None)`
 Get the form element plugin widget for the `plugin_uid` given.

Parameters

- **plugin_uid** (`str`) – UID of the plugin to get the widget for.
- **request** (`django.http.HttpRequest`) –
- **as_instance** (`bool`) –
- **theme** (`fobi.base.BaseTheme`) – Subclass of.

Return `BasePluginWidget` Subclass of.

`fobi.base.get_form_handler_plugin_widget(plugin_uid, request=None, as_instance=False, theme=None)`
 Get the form handler plugin widget for the `plugin_uid` given.

Parameters

- **plugin_uid** (`str`) – UID of the plugin to get the widget for.
- **request** (`django.http.HttpRequest`) –
- **as_instance** (`bool`) –
- **theme** (`fobi.base.BaseTheme`) – Subclass of.

Return `BasePluginWidget` Subclass of.

`fobi.base.get_form_wizard_handler_plugin_widget(plugin_uid, request=None, as_instance=False, theme=None)`

Get the form wizard handler plugin widget for the `plugin_uid` given.

Parameters

- **plugin_uid** (`str`) – UID of the plugin to get the widget for.
- **request** (`django.http.HttpRequest`) –
- **as_instance** (`bool`) –
- **theme** (`fobi.base.BaseTheme`) – Subclass of.

Return `BasePluginWidget` Subclass of.

`fobi.base.get_ordered_form_handlers()`

Get ordered form handler plugins.

Gets form handler plugins in the execution order as a sortable dictionary, which can be later on used to add real plugins to be executed.

Return `fobi.data_structures.SortableDict`

`fobi.base.get_ordered_form_wizard_handlers()`

Get ordered form wizard_handler plugins.

Gets form wizard handler plugins in the execution order as a sortable dictionary, which can be later on used to add real plugins to be executed.

Return `fobi.data_structures.SortableDict`

`fobi.base.get_plugin_widget(registry, plugin_uid, request=None, as_instance=False, theme=None)`

Get the plugin widget for the plugin_uid given.

Looks up in the registry provided.

Parameters

- **registry** (`fobi.base.BasePluginWidgetRegistry`) – Subclass of.
- **plugin_uid** (`str`) – UID of the plugin to get the widget for.
- **request** (`django.http.HttpRequest`) –
- **as_instance** (`bool`) –
- **theme** (`fobi.base.BaseTheme`) – Subclass of.

Return `BasePluginWidget` Subclass of.

`fobi.base.get_processed_form_data(form, form_element_entries)`

Gets processed form data.

Simply fires both `fobi.base.get_cleaned_data` and `fobi.base.get_field_name_to_label_map` functions and returns the result.

Parameters

- **form** (`django.forms.Form`) –
- **form_element_entries** (`iterable`) – Iterable of form element entries.

Return tuple

`fobi.base.get_processed_form_wizard_data(form_wizard, form_list, form_element_entries)`

Get processed form wizard data.

`fobi.base.get_registered_form_callbacks(stage=None)`

Get registered form callbacks for the stage given.

`fobi.base.get_registered_form_element_plugin_uids(flattern=True)`

Get registered form element plugin uids.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_element_plugins()`

Get registered form element plugins.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_element_plugins_grouped(sort_by_value=False)`

Get registered form element plugins grouped.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return dict

`fobi.base.get_registered_form_handler_plugin_uids(flattern=True)`

Get registered form handler plugin uids.

Gets a list of UIDs of registered form handler plugins. If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_handler_plugins(as_instances=False)`

Get registered form handler plugins.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_wizard_handler_plugin_uids(flattern=True)`

Get registered form handler plugin uids.

Gets a list of UIDs of registered form wizard handler plugins. If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_wizard_handler_plugins(as_instances=False)`

Get registered form handler wizard plugins.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_element_plugin_uids(flattern=True)`

Get registered custom field instance plugin uids.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_element_plugins()`

Get registered custom field instance plugins.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_element_plugins_grouped()`

Get registered custom field instance plugins grouped.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return dict

`fobi.base.get_registered_integration_form_handler_plugin_uids(flattern=True)`

Get registered integration form handler plugin uids.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_handler_plugins()`

Get registered integration form handler plugins.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_handler_plugins_grouped()`

Get registered integration form handler plugins grouped.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return dict

`fobi.base.get_registered_plugin_uids(registry, flatten=True, sort_items=True)`

Get a list of registered plugin uids as a list .

If not yet auto-discovered, auto-discovers them.

The `sort_items` is applied only if `flatten` is True.

Parameters

- **registry** –
- **flatten** (*bool*) –
- **sort_items** (*bool*) –

Return list

`fobi.base.get_registered_plugins(registry, as_instances=False, sort_items=True)`

Get registered plugins.

Get a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Parameters

- **registry** –
- **as_instances** (*bool*) –
- **sort_items** (*bool*) –

Return list

`fobi.base.get_registered_theme_uids(flatten=True)`

Get registered theme uids.

Gets a list of registered themes in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_themes()`

Get registered themes.

Gets a list of registered themes in form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_theme(request=None, theme_uid=None, as_instance=False)`

Get theme.

Gets the theme by `theme_uid` given. If left empty, takes the default one chosen in `settings` module.

Raises a `fobi.exceptions.ThemeDoesNotExist` when no default layout could be found.

Parameters

- **request** (`django.http.HttpRequest`) –
- **theme_uid** (`int`) –
- **as_instance** (`bool`) –

Return `fobi.base.BaseTheme` Subclass of `fobi.base.BaseTheme`.

`fobi.base.run_form_handlers(form_entry, request, form, form_element_entries=None)`

Run form handlers.

Parameters

- **form_entry** (`fobi.models.FormEntry`) –
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** (`iterable`) –

Return tuple List of success responses, list of error responses

`fobi.base.run_form_wizard_handlers(form_wizard_entry, request, form_list, form_wizard, form_element_entries=None)`

Run form wizard handlers.

Parameters

- **form_wizard_entry** (`fobi.models.FormWizardEntry`) –
- **request** (`django.http.HttpRequest`) –
- **form_list** (`list`) – List of `django.forms.Form` objects.
- **form_wizard** (`fobi.wizard.views.dynamic.DynamicWizardView`) – The form wizard view object.
- **form_element_entries** (`iterable`) – Iterable of `fobi.base.FormElementEntry` objects.

Return tuple List of success responses, list of error responses

`fobi.base.submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)`

Submit plugin form data for all plugins.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** (`iterable`) –

`fobi.base.validate_form_element_plugin_uid(plugin_uid)`

Validate the form element plugin uid.

Parameters `plugin_uid` (`string`) –

Return bool

`fobi.base.validate_form_handler_plugin_uid(plugin_uid)`
 Validate the plugin uid.

Parameters `plugin_uid (string)` –

Return bool

`fobi.base.validate_form_wizard_handler_plugin_uid(plugin_uid)`
 Validate the plugin uid.

Parameters `plugin_uid (string)` –

Return bool

`fobi.base.validate_integration_form_element_plugin_uid(plugin_uid)`
 Validate the custom field instance plugin uid.

Parameters `plugin_uid (string)` –

Return bool

`fobi.base.validate_integration_form_handler_plugin_uid(plugin_uid)`
 Validate the integration form handler plugin uid.

Parameters `plugin_uid (string)` –

Return bool

`fobi.base.validate_theme_uid(plugin_uid)`
 Validate the theme uid.

Parameters `plugin_uid (string)` –

Return bool

48.4.7 fobi.conf module

`fobi.conf.get_setting(setting, override=None)`
 Get setting.

Get a setting from *fobi* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

48.4.8 fobi.constants module

48.4.9 fobi.context_processors module

`fobi.context_processors.dynamic_values(request)`

Dynamic values exposed to public forms.

`fobi.context_processors.form_importers(request)`

Form importers.

`fobi.context_processors.theme(request)`

Get active theme.

Parameters `request` (`django.http.HttpRequest`) –

Return `fobi.base.BaseTheme` Instance of `fobi.base.BaseTheme`.

48.4.10 fobi.data_structures module

`class fobi.data_structures.SortableDict(*args, **kwargs)`

Bases: `dict`

`SortableDict`.

A dictionary that keeps its keys in the order in which they're inserted. Very similar to (and partly based on) `SortedDict` of the Django, but has several additional methods implemented, such as: `insert_before_key` and `insert_after_key`.

clear()

Clear.

copy()

Returns a copy of this object.

insert(index, key, value)

Inserts the key, value pair before the item with the given index.

insert_after_key(target_key, key, value, fail_silently=True)

Insert the {key: value} after the `target_key`.

Parameters

- **target_key** (*immutable*) –
- **key** (*immutable*) –
- **value** (*mutable*) –
- **fail_silently** (*boolean*) –
- **offset** (*int*) –

Return `bool`

insert_before_key(target_key, key, value, fail_silently=True, offset=0)

Insert the {key: value} before the `target_key`.

Parameters

- **target_key** (*immutable*) –
- **key** (*immutable*) –
- **value** (*mutable*) –

- **fail_silently** (*boolean*) –
- **offset** (*int*) –

Return bool

items()

Iter items (internal method).

keys() → a set-like object providing a view on D's keys

move_after_key(*source_key*, *target_key*, *fail_silently=True*)

Move the {key: value} after the given *source_key*.

Parameters

- **source_key** (*immutable*) –
- **target_key** (*immutable*) –
- **fail_silently** (*boolean*) –

Return bool

move_before_key(*source_key*, *target_key*, *fail_silently=True*, *offset=0*)

Move the {key: value} before the given *source_key*.

Parameters

- **source_key** (*immutable*) –
- **target_key** (*immutable*) –
- **fail_silently** (*boolean*) –
- **offset** (*int*) –

Return bool

pop(*key*, **args*)

Pop.

popitem()

Pop item.

setdefault(*key*, *default*)

Set default.

update(*dict_*)

Update.

value_for_index(*index*)

Returns the value of the item at the given zero-based index.

values()

Iter values (internal method).

48.4.11 fobi.decorators module

`fobi.decorators.all_permissions_required(perms, login_url=None, raise_exception=False)`
Check for the permissions given based on SATISFY_ALL strategy chosen.

Example

```
>>> @login_required
>>> @all_permissions_required([
>>>     'fobi.add_formentry',
>>>     'fobi.change_formentry',
>>>     'fobi.delete_formentry',
>>>     'fobi.add_formelemententry',
>>>     'fobi.change_formelemententry',
>>>     'fobi.delete_formelemententry',
>>> ])
>>> def edit_dashboard(request):
>>>     # your code
```

`fobi.decorators.any_permission_required(perms, login_url=None, raise_exception=False)`
Check for the permissions given based on SATISFY_ANY strategy chosen.

Example

```
>>> @login_required
>>> @any_permission_required([
>>>     'fobi.add_formentry',
>>>     'fobi.change_formentry',
>>>     'fobi.delete_formentry',
>>>     'fobi.add_formelemententry',
>>>     'fobi.change_formelemententry',
>>>     'fobi.delete_formelemententry',
>>> ])
>>> def edit_dashboard(request):
>>>     # your code
```

`fobi.decorators.permissions_required(perms, satisfy='all', login_url=None, raise_exception=False)`
Check for the permissions given based on the strategy chosen.

Parameters

- **perms** (*iterable*) –
- **satisfy** (*string*) – Allowed values are “all” and “any”.
- **login_url** (*string*) –
- **raise_exception** (*bool*) – If set to True, the `PermissionDenied` exception is raised on failures.

Return bool

Example

```
>>> @login_required
>>> @permissions_required(satisfy='any', perms=[
>>>     'fobi.add_formentry',
>>>     'fobi.change_formentry',
```

(continues on next page)

(continued from previous page)

```
>>> 'fobi.delete_formentry',
>>> 'fobi.add_formelemententry',
>>> 'fobi.change_formelemententry',
>>> 'fobi.delete_formelemententry',
>>> ])
>>> def edit_dashboard(request):
>>>     # your code
```

48.4.12 fobi.defaults module

48.4.13 fobi.discover module

`fobi.discover.autodiscover()`

Auto-discovers files that should be found by fobi.

48.4.14 fobi.dynamic module

`fobi.dynamic.assemble_form_class(form_entry, base_class=<class 'django.forms.forms.BaseForm'>, request=None, origin=None, origin_kwargs_update_func=None, origin_return_func=None, form_element_entries=None, get_form_field_instances_kwargs={})`

Assemble a form class by given entry.

Parameters

- **form_entry** –
- **base_class** –
- **request** (*django.http.HttpRequest*) –
- **origin** (*string*) –
- **origin_kwargs_update_func** (*callable*) –
- **origin_return_func** (*callable*) –
- **form_element_entries** (*iterable*) – If given, used instead of `form_entry.formelemententry_set.all` (no additional database hit).
- **get_form_field_instances_kwargs** (*dict*) – To be passed as ****kwargs** to the **method: 'get_form_field_instances_kwargs'**.

`fobi.dynamic.assemble_form_wizard_class(form_wizard_entry, base_class=<class 'formtools.wizard.views.SessionWizardView'>, request=None, origin=None, origin_kwargs_update_func=None, origin_return_func=None, form_wizard_form_entries=None, template_name=None)`

Assemble form wizard class.

Parameters

- **form_wizard_entry** –
- **base_class** –
- **request** –

- `origin` –
- `origin_kwargs_update_func` –
- `origin_return_func` –
- `form_wizard_form_entries` –
- `template_name` –

Returns

48.4.15 fobi.exceptions module

exception `fobi.exceptions.BaseException`

Bases: `Exception`

Base exception.

exception `fobi.exceptions.DoesNotExist`

Bases: `fobi.exceptions.BaseException`

Raised when something does not exist.

exception `fobi.exceptions.FormCallbackError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form callback error occurs.

exception `fobi.exceptions.FormElementPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

Raised when no form element plugin with given uid can be found.

exception `fobi.exceptions.FormElementPluginError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form element plugin error occurs.

exception `fobi.exceptions.FormHandlerPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

Raised when no form handler plugin with given uid can be found.

exception `fobi.exceptions.FormHandlerPluginError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form handler plugin error occurs.

exception `fobi.exceptions.FormPluginError`

Bases: `fobi.exceptions.BaseException`

Base error for form elements and handlers.

exception `fobi.exceptions.FormWizardHandlerPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

FormWizardHandlerPlugin does not exist.

Raised when no form wizard handler plugin with given uid can be found.

exception `fobi.exceptions.ImproperlyConfigured`

Bases: `fobi.exceptions.BaseException`

Improperly configured.

Exception raised when developer didn't configure/write the code properly.

exception `fobi.exceptions.IntegrationFormElementPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

If integration form element plugin with given uid can't be found.

exception `fobi.exceptions.IntegrationFormHandlerPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

If integration form handler plugin with given uid can't be found.

exception `fobi.exceptions.InvalidRegistryItemType`

Bases: `ValueError`, `fobi.exceptions.BaseException`

Invalid registry item type.

Raised when an attempt is made to register an item in the registry which does not have a proper type.

exception `fobi.exceptions.NoDefaultThemeSet`

Bases: `fobi.exceptions.ImproperlyConfigured`

Raised when no active theme is chosen.

exception `fobi.exceptions.PluginDoesNotExist`

Bases: `fobi.exceptions.DoesNotExist`

Raised when no plugin with given uid can be found.

exception `fobi.exceptions.ThemeDoesNotExist`

Bases: `fobi.exceptions.DoesNotExist`

Raised when no theme with given uid can be found.

48.4.16 fobi.form_importers module

class `fobi.form_importers.BaseFormImporter`(*form_entry_cls, form_element_entry_cls, form_properties=None, form_data=None*)

Bases: `object`

Base importer.

description = `None`

extract_field_properties(*field_data*)

Extract field properties.

field_properties_mapping = `None`

field_type_prop_name = `None`

fields_mapping = `None`

get_form_data()

Get form data.

get_template_names()

Get template names.

get_wizard(*request, *args, **kwargs*)

Get wizard.

import_data(*form_properties, form_data*)

Import data.

```
name = None
position_prop_name = None
templates = None
uid = None
wizard = None

class fobi.form_importers.FormImporterPluginRegistry
    Bases: fobi.base.BaseRegistry
    Form importer plugins registry.

    type
        alias of fobi.form_importers.BaseFormImporter

fobi.form_importers.ensure_autodiscover()
    Ensure that form importer plugins are auto-discovered.

fobi.form_importers.get_form_importer_plugin_uids()
    Get form importer plugin uids.

fobi.form_importers.get_form_importer_plugin_urls()
    Gets the form importer plugin URLs as a list of tuples.
```

48.4.17 fobi.form_utils module

```
class fobi.form_utils.ErrorDict
    Bases: django.forms.utils.ErrorDict
    A better ErrorDict.

    as_text()
        As text.

class fobi.form_utils.ErrorList(initlist=None, error_class=None)
    Bases: django.forms.utils.ErrorList
    A better ErrorList.

    as_text()
        As text.
```

48.4.18 fobi.forms module

```
class fobi.forms.BulkChangeFormElementPluginsForm(*args, **kwargs)
    Bases: fobi.forms.BaseBulkChangePluginsForm
    Bulk change form element plugins form.

    class Meta
        Bases: object
        Meta class.

        fields = ['groups', 'groups_action', 'users', 'users_action']

    model
        alias of fobi.models.FormElement
```

```
base_fields = {'groups': <django.forms.models.ModelMultipleChoiceField object>,
'groups_action': <django.forms.fields.ChoiceField object>, 'selected_plugins':
<django.forms.fields.CharField object>, 'users':
<django.forms.models.ModelMultipleChoiceField object>, 'users_action':
<django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'groups_action': <django.forms.fields.ChoiceField object>,
'selected_plugins': <django.forms.fields.CharField object>, 'users_action':
<django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

```
class fobi.forms.BulkChangeFormHandlerPluginsForm(*args, **kwargs)
```

Bases: `fobi.forms.BaseBulkChangePluginsForm`

Bulk change form handler plugins form.

class Meta

Bases: `object`

Meta class.

```
fields = ['groups', 'groups_action', 'users', 'users_action']
```

model

alias of `fobi.models.FormHandler`

```
base_fields = {'groups': <django.forms.models.ModelMultipleChoiceField object>,
'groups_action': <django.forms.fields.ChoiceField object>, 'selected_plugins':
<django.forms.fields.CharField object>, 'users':
<django.forms.models.ModelMultipleChoiceField object>, 'users_action':
<django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'groups_action': <django.forms.fields.ChoiceField object>,
'selected_plugins': <django.forms.fields.CharField object>, 'users_action':
<django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

```
class fobi.forms.BulkChangeFormWizardHandlerPluginsForm(*args, **kwargs)
```

Bases: `fobi.forms.BaseBulkChangePluginsForm`

Bulk change form wizard handler plugins form.

class Meta

Bases: `object`

Meta class.

```
fields = ['groups', 'groups_action', 'users', 'users_action']
```

model

alias of `fobi.models.FormWizardHandler`

```
base_fields = {'groups': <django.forms.models.ModelMultipleChoiceField object>,
'groups_action': <django.forms.fields.ChoiceField object>, 'selected_plugins':
<django.forms.fields.CharField object>, 'users':
<django.forms.models.ModelMultipleChoiceField object>, 'users_action':
<django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'groups_action': <django.forms.fields.ChoiceField object>,
'selected_plugins': <django.forms.fields.CharField object>, 'users_action':
<django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

```
class fobi.forms.FormElementEntryForm(data=None, files=None, auto_id='id_%s', prefix=None,
initial=None, error_class=<class 'django.forms.utils.ErrorList'>,
label_suffix=None, empty_permitted=False, instance=None,
use_required_attribute=None, renderer=None)
```

Bases: `django.forms.models.ModelForm`

FormElementEntry form.

class Meta

Bases: `object`

Meta class.

```
fields = ('form_entry', 'plugin_data', 'plugin_uid', 'position')
```

model

alias of `fobi.models.FormElementEntry`

```
base_fields = {'form_entry': <django.forms.models.ModelChoiceField object>,
'plugin_data': <django.forms.fields.CharField object>, 'plugin_uid':
<django.forms.fields.ChoiceField object>, 'position':
<django.forms.fields.IntegerField object>}
```

```
declared_fields = {'plugin_uid': <django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

`fobi.forms.FormElementEntryFormSet`

alias of `django.forms.formsets.FormElementEntryFormFormSet`

```
class fobi.forms.FormEntryForm(*args, **kwargs)
```

Bases: `django.forms.models.ModelForm`

Form for `fobi.models.FormEntry` model.

class Meta

Bases: `object`

Meta class.

```
fields = ('name', 'title', 'is_public', 'active_date_from', 'active_date_to',
'inactive_page_title', 'inactive_page_message', 'success_page_title',
'success_page_message', 'action')
```

model

alias of `fobi.models.FormEntry`

```

base_fields = {'action': <django.forms.fields.CharField object>,
'active_date_from': <django.forms.fields.DateTimeField object>, 'active_date_to':
<django.forms.fields.DateTimeField object>, 'inactive_page_message':
<django.forms.fields.CharField object>, 'inactive_page_title':
<django.forms.fields.CharField object>, 'is_public':
<django.forms.fields.BooleanField object>, 'name': <django.forms.fields.CharField
object>, 'success_page_message': <django.forms.fields.CharField object>,
'success_page_title': <django.forms.fields.CharField object>, 'title':
<django.forms.fields.CharField object>}

clean_action()
    Validate the action (URL).

    Checks if URL exists.

declared_fields = {}

property media
    Return all media required to render the widgets on this form.
class fobi.forms.ModelFormEntryForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    Form for fobi.models.ModelFormEntry model.

    class Meta
        Bases: object

        Meta class.

        fields = ('name',)

        model
            alias of fobi.models.ModelFormEntry

    base_fields = {'name': <django.forms.fields.CharField object>}

    declared_fields = {}

    property media
        Return all media required to render the widgets on this form.
class fobi.forms.ModelFormHandlerEntryForm(data=None, files=None, auto_id='id_%s', prefix=None,
initial=None, error_class=<class 'django.forms.utils.ErrorList'>,
label_suffix=None, empty_permitted=False, instance=None,
use_required_attribute=None, renderer=None)

    Bases: django.forms.models.ModelForm

    FormHandlerEntry form.

    class Meta
        Bases: object

        Meta class.

        fields = ('form_entry', 'plugin_data', 'plugin_uid')

        model
            alias of fobi.models.ModelFormHandlerEntry

    base_fields = {'form_entry': <django.forms.models.ModelChoiceField object>,
'plugin_data': <django.forms.fields.CharField object>, 'plugin_uid':
<django.forms.fields.ChoiceField object>}

```

```
declared_fields = {'plugin_uid': <django.forms.fields.ChoiceField object>}
```

property media

Return all media required to render the widgets on this form.

```
class fobi.forms.FormHandlerForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None,
                                error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None,
                                empty_permitted=False, instance=None, use_required_attribute=None,
                                renderer=None)
```

Bases: `django.forms.models.ModelForm`

FormHandler form.

class Meta

Bases: `object`

Meta class.

```
fields = ('users', 'groups')
```

model

alias of `fobi.models.FormHandler`

```
base_fields = {'groups': <django.forms.models.ModelMultipleChoiceField object>,
               'users': <django.forms.models.ModelMultipleChoiceField object>}
```

```
declared_fields = {}
```

property media

Return all media required to render the widgets on this form.

```
class fobi.forms.FormWizardEntryForm(*args, **kwargs)
```

Bases: `django.forms.models.ModelForm`

Form for `fobi.models.FormWizardEntry` model.

class Meta

Bases: `object`

Meta class.

```
fields = ('name', 'title', 'is_public', 'success_page_title',
          'success_page_message', 'show_all_navigation_buttons')
```

model

alias of `fobi.models.FormWizardEntry`

```
base_fields = {'is_public': <django.forms.fields.BooleanField object>, 'name':
               <django.forms.fields.CharField object>, 'show_all_navigation_buttons':
               <django.forms.fields.BooleanField object>, 'success_page_message':
               <django.forms.fields.CharField object>, 'success_page_title':
               <django.forms.fields.CharField object>, 'title': <django.forms.fields.CharField
               object>}
```

```
declared_fields = {}
```

property media

Return all media required to render the widgets on this form.


```
class fobi.forms.FormWizardFormEntryForm(data=None, files=None, auto_id='id_%s', prefix=None,
                                         initial=None, error_class=<class
                                         'django.forms.utils.ErrorList'>, label_suffix=None,
                                         empty_permitted=False, instance=None,
                                         use_required_attribute=None, renderer=None)
```

Bases: django.forms.models.ModelForm

FormWizardFormEntryForm form.

```
class Meta
```

Bases: object

Meta class.

```
fields = ('form_wizard_entry', 'form_entry')
```

```
model
```

alias of [fobi.models.FormWizardFormEntry](#)

```
base_fields = {'form_entry': <django.forms.models.ModelChoiceField object>,
                'form_wizard_entry': <django.forms.models.ModelChoiceField object>}
```

```
declared_fields = {}
```

```
property media
```

Return all media required to render the widgets on this form.

```
fobi.forms.FormWizardFormEntryFormSet
```

alias of django.forms.formsets.FormWizardFormEntryFormFormSet

```
class fobi.forms.FormWizardHandlerEntryForm(data=None, files=None, auto_id='id_%s', prefix=None,
                                             initial=None, error_class=<class
                                             'django.forms.utils.ErrorList'>, label_suffix=None,
                                             empty_permitted=False, instance=None,
                                             use_required_attribute=None, renderer=None)
```

Bases: django.forms.models.ModelForm

FormWizardHandlerEntry form.

```
class Meta
```

Bases: object

Meta class.

```
fields = ('form_wizard_entry', 'plugin_data', 'plugin_uid')
```

```
model
```

alias of [fobi.models.FormWizardHandlerEntry](#)

```
base_fields = {'form_wizard_entry': <django.forms.models.ModelChoiceField object>,
                'plugin_data': <django.forms.fields.CharField object>, 'plugin_uid':
                <django.forms.fields.ChoiceField object>}
```

```
declared_fields = {'plugin_uid': <django.forms.fields.ChoiceField object>}
```

```
property media
```

Return all media required to render the widgets on this form.

```
class fobi.forms.ImportFormEntryForm(data=None, files=None, auto_id='id_%s', prefix=None,
                                       initial=None, error_class=<class 'django.forms.utils.ErrorList'>,
                                       label_suffix=None, empty_permitted=False, field_order=None,
                                       use_required_attribute=None, renderer=None)
```

Bases: django.forms.forms.Form

Import form entry form.

```
base_fields = {'file': <django.forms.fields.FileField object>}
```

```
declared_fields = {'file': <django.forms.fields.FileField object>}
```

property media

Return all media required to render the widgets on this form.

```
class fobi.forms.ImportFormWizardEntryForm(data=None, files=None, auto_id='id_%s', prefix=None,
                                             initial=None, error_class=<class
                                             'django.forms.utils.ErrorList'>, label_suffix=None,
                                             empty_permitted=False, field_order=None,
                                             use_required_attribute=None, renderer=None)
```

Bases: *fobi.forms.ImportFormEntryForm*

Import form entry wizard form.

```
base_fields = {'file': <django.forms.fields.FileField object>}
```

```
declared_fields = {'file': <django.forms.fields.FileField object>}
```

property media

Return all media required to render the widgets on this form.

48.4.19 fobi.helpers module

Helpers module. This module can be safely imported from any fobi (sub)module, since it never imports from any of the fobi (sub)modules (except for the *fobi.constants* and *fobi.exceptions* modules).

```
class fobi.helpers.JSONDataExporter(data, filename)
```

Bases: object

Exporting the data into JSON.

export()

Export.

export_to_json()

Export data to JSON.

```
class fobi.helpers.StrippedRequest(request)
```

Bases: object

Stripped request object.

property META

Request meta stripped down.

A standard Python dictionary containing all available HTTP headers. Available headers depend on the client and server, but here are some examples:

- HTTP_ACCEPT_ENCODING: Acceptable encodings for the response.
- HTTP_ACCEPT_LANGUAGE: Acceptable languages for the response.
- HTTP_HOST: The HTTP Host header sent by the client.
- HTTP_REFERER: The referring page, if any.
- HTTP_USER_AGENT: The clients user-agent string.
- QUERY_STRING: The query string, as a single (unparsed) string.
- REMOTE_ADDR: The IP address of the client.

get_full_path()

Returns the path, plus an appended query string, if applicable.

is_ajax()

Is ajax?

Returns True if the request was made via an XMLHttpRequest, by checking the HTTP_X_REQUESTED_WITH header for the string 'XMLHttpRequest'.

is_secure()

Is secure.

Returns True if the request is secure; that is, if it was made with HTTPS.

property path

Path.

A string representing the full path to the requested page, not including the scheme or domain.

class fobi.helpers.StrippedUser(*user*)

Bases: object

Stripped user object.

property email

Email.

get_full_name()

Get full name.

get_short_name()

Get short name.

get_username()

Get username.

is_anonymous()

Is anonymous.

fobi.helpers.admin_change_url(*app_label, module_name, object_id, extra_path="", url_title=None*)

Gets an admin change URL for the object given.

Parameters

- **app_label** (*str*) –
- **module_name** (*str*) –
- **object_id** (*int*) –
- **extra_path** (*str*) –
- **url_title** (*str*) – If given, an HTML a tag is returned with *url_title* as the tag title. If left to None just the URL string is returned.

Return str**fobi.helpers.clean_dict(*source, keys=[], values=[]*)**

Removes given keys and values from dictionary.

Parameters

- **source** (*dict*) –
- **keys** (*iterable*) –
- **values** (*iterable*) –

Return dict

`fobi.helpers.clone_file(upload_dir, source_filename, relative_path=True)`
 Clones the file.

Parameters `source_filename` (*string*) – Source filename.

Return string Filename of the cloned file.

`fobi.helpers.combine_dicts(headers, data)`
 Combine dicts.

Takes two dictionaries, assuming one contains a mapping keys to titles and another keys to data. Joins as string and returns a result dict.

`fobi.helpers.delete_file(image_file)`
 Delete file from disc.

`fobi.helpers.do_slugify(val)`
 Slugify.

`fobi.helpers.ensure_unique_filename(destination)`
 Makes sure filenames are never overwritten.

Parameters `destination` (*string*) –

Return string

`fobi.helpers.extract_file_path(name)`
 Extracts the file path.

Parameters `name` (*string*) –

Return string

`fobi.helpers.flatatt_inverse_quotes(attrs)`
 Convert a dictionary of attributes to a single string.

The returned string will contain a leading space followed by key="value", XML-style pairs. In the case of a boolean value, the key will appear without a value. It is assumed that the keys do not need to be XML-escaped. If the passed dictionary is empty, then return an empty string.

The result is passed through 'mark_safe' (by way of 'format_html_join').

`fobi.helpers.get_app_label_and_model_name(path)`
 Gets app_label and model_name from the path given.

Parameters `path` (*str*) – Dotted path to the model (without ".model", as stored in the Django *ContentType* model.

Return tuple app_label, model_name

`fobi.helpers.get_form_element_entries_for_form_wizard_entry(form_wizard_entry)`
 Get form element entries for the form wizard entry.

`fobi.helpers.get_ignorable_form_values()`
 Get ignorable for form values.

Gets an iterable of form values to ignore.

Return iterable

`fobi.helpers.get_model_name_for_object(obj)`
 Get model name for object.

Django version agnostic.

`fobi.helpers.get_registered_models(ignore=[])`

Gets registered models as list.

Parameters `ignore` (*iterable*) – Ignore the following content types (should be in `app_label.model` format (example `auth.User`)).

Return list

`fobi.helpers.get_select_field_choices(raw_choices_data, key_type=None, value_type=None, fail_silently=True)`

Get select field choices.

Used in radio, select and other choice based fields.

Parameters

- `raw_choices_data` (*str*) –
- `key_type` (*type*) –
- `value_type` (*type*) –
- `fail_silently` (*bool*) –

Return list

`fobi.helpers.get_wizard_form_field_value_from_post(request, wizard_view_name, form_key, field_name, fail_silently=True)`

Get wizard form field value from POST.

This is what we could have:

```
>>> request.POST
>>> {
>>>     'csrfmiddlewaretoken': ['kEprTL218a8HNcC02QefNNnF'],
>>>     'slider-form-test_slider': ['14'],
>>>     'form_wizard_view-current_step': ['slider-form'],
>>>     'slider-form-test_email': ['user@example.com']
>>> }
```

Note, that we know nothing about the types here, type conversion should be done manually. The values returned are strings always.

Parameters

- `request` (*django.http.HttpRequest*) –
- `wizard_view_name` (*str*) –
- `form_key` (*str*) – Typically, this would be the step name (form slug).
- `field_name` (*str*) – Field name.
- `fail_silently` (*bool*) – If set to True, no errors raised.

Return str Since everything in session is stored as string.

`fobi.helpers.get_wizard_form_field_value_from_request(request, wizard_view_name, form_key, field_name, fail_silently=True, session_priority=False)`

Get wizard form field value from request.

Note, that we know nothing about the types here, type conversion should be done manually. The values returned are strings always.

Parameters

- **request** (*django.http.HttpRequest*) –
- **wizard_view_name** (*str*) –
- **form_key** (*str*) – Typically, this would be the step name (form slug).
- **field_name** (*str*) – Field name.
- **fail_silently** (*bool*) – If set to True, no errors raised.
- **session_priority** (*bool*) – If set to True, first try to read from session.

Return str Since everything in session is stored as string.

`fobi.helpers.get_wizard_form_field_value_from_session(request, wizard_view_name, form_key, field_name, fail_silently=True)`

Get wizard form field value from session.

This is what we could have:

```
>>> request.session['wizard_form_wizard_view']['step_data']
>>> {
>>>     'slider-form': {
>>>         'csrfmiddlewaretoken': ['DhINThGTgQ50e2lDnGG4nYrG0a'],
>>>         'slider-form-test_slider': ['14'],
>>>         'form_wizard_view-current_step': ['slider-form'],
>>>         'slider-form-test_email': ['user@example.com']
>>>     }
>>> }
```

Note, that we know nothing about the types here, type conversion should be done manually. The values returned are strings always.

Parameters

- **request** (*django.http.HttpRequest*) –
- **wizard_view_name** (*str*) –
- **form_key** (*str*) – Typically, this would be the step name (form slug).
- **field_name** (*str*) – Field name.
- **fail_silently** (*bool*) – If set to True, no errors raised.

Return str Since everything in session is stored as string.

`fobi.helpers.handle_uploaded_file(upload_dir, image_file)`

Handle uploaded files.

Parameters **image_file** (*django.core.files.uploadedfile.InMemoryUploadedFile*) –

Return string Path to the image (relative).

`fobi.helpers.iterable_to_dict(items, key_attr_name)`

Converts iterable of certain objects to dict.

Parameters

- **items** (*iterable*) –
- **key_attr_name** (*string*) – Attribute to use as a dictionary key.

Return dict

`fobi.helpers.lists_overlap(sub, main)`

Check whether lists overlap.

`fobi.helpers.map_field_name_to_label(form)`

Takes a form and creates label to field name map.

Parameters `form` (`django.forms.Form`) – Instance of `django.forms.Form`.

Return dict

`fobi.helpers.safe_text(text)`

Safe text (encode).

Return str

`fobi.helpers.two_dicts_to_string(headers, data, html_element='p')`

Two dicts to string.

Takes two dictionaries, assuming one contains a mapping keys to titles and another keys to data. Joins as string and returns wrapped into HTML “p” tag.

`fobi.helpers.uniquify_sequence(sequence)`

Uniqify sequence.

Makes sure items in the given sequence are unique, having the original order preserved.

Parameters `sequence` (`iterable`) –

Return list

`fobi.helpers.update_plugin_data(entry, request=None)`

Update plugin data.

Update plugin data of a given entry.

`fobi.helpers.validate_initial_for_choices(plugin_form, field_name_choices='choices', field_name_initial='initial')`

Validate init for choices. Validates the initial value for the choices given.

Parameters

- `plugin_form` (`fobi.base.BaseFormFieldPluginForm`) –
- `field_name_choices` (`str`) –
- `field_name_initial` (`str`) –

Return str

`fobi.helpers.validate_initial_for_multiple_choices(plugin_form, field_name_choices='choices', field_name_initial='initial')`

Validates the initial value for the multiple choices given.

Parameters

- `plugin_form` (`fobi.base.BaseFormFieldPluginForm`) –
- `field_name_choices` (`str`) –
- `field_name_initial` (`str`) –

Return str

`fobi.helpers.validate_submit_value_as(value)`

Validates the `SUBMIT_AS_VALUE`.

Parameters `value` (`str`) –

48.4.20 fobi.models module

class `fobi.models.AbstractFormWizardPluginEntry(*args, **kwargs)`

Bases: `fobi.models.BaseAbstractPluginEntry`

Abstract form wizard plugin entry.

Properties

- `form_entry` (`fobi.models.FormWizardEntry`): FormWizard to which the plugin belongs to.
- `plugin_uid` (str): Plugin UID.
- `plugin_data` (str): JSON formatted string with plugin data.

class `Meta`

Bases: `object`

Meta class.

abstract = `False`

property `entry_user`

Get user.

form_wizard_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

form_wizard_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class `fobi.models.AbstractPluginEntry(*args, **kwargs)`

Bases: `fobi.models.BaseAbstractPluginEntry`

Abstract plugin entry.

Properties

- `form_entry` (`fobi.models.FormEntry`): Form to which the field plugin belongs to.
- `plugin_uid` (str): Plugin UID.
- `plugin_data` (str): JSON formatted string with plugin data.

class `Meta`

Bases: `object`

Meta class.

abstract = `False`

property `entry_user`

Get user.

form_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

form_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class fobi.models.**AbstractPluginModel**(*args, **kwargs)

Bases: django.db.models.base.Model

Abstract plugin model.

Used when fobi.settings.RESTRICT_PLUGIN_ACCESS is set to True.

Properties

- *plugin_uid* (str): Plugin UID.
- *users* (django.contrib.auth.models.User): White list of the users allowed to use the plugin.
- *groups* (django.contrib.auth.models.Group): White list of the user groups allowed to use the plugin.

class Meta

Bases: object

Meta class.

abstract = False

get_registered_plugins()

Get registered plugins.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

Pizza.toppings and Topping.pizzas are ManyToManyDescriptor instances.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

groups_list()

Groups list.

Flat list (comma separated string) of groups allowed to use the plugin. Used in Django admin.

Return string**plugin_uid_admin()**

Plugin uid admin.

Mainly used in admin.

plugin_uid_code()

Plugin uid code.

Mainly used in admin.

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

users_list()

Users list.

Flat list (comma separated string) of users allowed to use the plugin. Used in Django admin.

Return string

class `fobi.models.BaseAbstractPluginEntry(*args, **kwargs)`

Bases: `django.db.models.base.Model`

Base for `AbstractPluginEntry`.

Properties

- `plugin_data` (str): JSON formatted string with plugin data.

class `Meta`

Bases: `object`

Meta class.

abstract = False

property `entry_user`

Get user from the parent container.

get_plugin(`fetch_related_data=False, request=None`)

Get plugin.

Gets the plugin class (by `plugin_uid` property), makes an instance of it, serves the data stored in `plugin_data` field (if available). Once all is done, plugin is ready to be rendered.

Parameters `fetch_related_data` (*bool*) – When set to True, plugin is told to re-fetch all related data (stored in models or other sources).

Return `fobi.base.BasePlugin` Subclass of `fobi.base.BasePlugin`.

get_registered_plugins()

Get registered plugins.

get_registry()

Get registry.

plugin_data

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

plugin_name()
Plugin name.

plugin_uid_code()
Plugin uid code.

Mainly used in admin.

class fobi.models.FormElement(*args, **kwargs)

Bases: [fobi.models.AbstractPluginModel](#)

Form element.

Form field plugin. Used when `fobi.settings.RESTRICT_PLUGIN_ACCESS` is set to `True`.

Properties

- *plugin_uid* (str): Plugin UID.
- *users* (django.contrib.auth.models.User): White list of the users allowed to use the form element plugin.
- *groups* (django.contrib.auth.models.Group): White list of the user groups allowed to use the form element plugin.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

get_registered_plugins()

Add choices.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class `fobi.models.FormElementEntry(*args, **kwargs)`

Bases: `fobi.models.AbstractPluginEntry`

Form field entry.

Properties

- *form* (`fobi.models.FormEntry`): Form to which the field plugin belongs to.
- *plugin_uid* (str): Plugin UID.
- *plugin_data* (str): JSON formatted string with plugin data.
- *form_fieldset_entry*: Fieldset.
- *position* (int): Entry position.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`form_entry`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

`form_fieldset_entry`

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

`form_fieldset_entry_id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`get_registered_plugins()`

Gets registered plugins.

`get_registry()`

Get registry.

`id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

position

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class fobi.models.**FormEntry**(*args, **kwargs)

Bases: django.db.models.base.Model

Form entry.

exception **DoesNotExist**

Bases: django.core.exceptions.ObjectDoesNotExist

exception **MultipleObjectsReturned**

Bases: django.core.exceptions.MultipleObjectsReturned

action

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

active_date_from

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

active_date_to

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

formelemententry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

formfieldsetentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

formhandlerentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

formwizardformentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_absolute_url()

Get absolute URL.

Absolute URL, which goes to the form-entry view view page.

Return string

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

inactive_page_message

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

inactive_page_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

property is_active**is_cloneable**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_public

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

savedformdataentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

slug

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

success_page_message

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

success_page_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class `fobi.models.ModelFormEntry(*args, **kwargs)`

Bases: `django.db.models.base.Model`

Form fieldset entry.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

form_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

formelemententry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_repeatable

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

class fobi.models.**FormHandler**(*args, **kwargs)

Bases: [fobi.models.AbstractPluginModel](#)

Form handler plugin. Used when `fobi.settings.RESTRICT_PLUGIN_ACCESS` is set to True.

Properties

- *plugin_uid* (str): Plugin UID.
- *users* (django.contrib.auth.models.User): White list of the users allowed to use the form handler plugin.
- *groups* (django.contrib.auth.models.Group): White list of the user groups allowed to use the form handler plugin.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

get_registered_plugins()

Add choices.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class `fobi.models.FormHandlerEntry(*args, **kwargs)`

Bases: `fobi.models.AbstractPluginEntry`

Form handler entry.

Properties

- `form_entry` (`fobi.models.FormEntry`): Form to which the handler plugin belongs to.
- `plugin_uid` (str): Plugin UID.
- `plugin_data` (str): JSON formatted string with plugin data.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

get_registered_plugins()

Gets registered plugins.

get_registry()

Get registry.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class fobi.models.FormWizardEntry(*args, **kwargs)

Bases: `django.db.models.base.Model`

Form wizard entry.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

formwizardformentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

formwizardhandlerentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_absolute_url()

Get absolute URL.

Absolute URL, which goes to the form-wizard view view.

Return string

get_wizard_type_display(**,field=<django.db.models.fields.CharField: wizard_type>*)

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_cloneable

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

is_public

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

name

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

savedformwizarddataentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

show_all_navigation_buttons

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

slug

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

success_page_message

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

success_page_title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

title

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

updated

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

user

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

user_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

wizard_type

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class fobi.models.**FormWizardFormEntry**(*args, **kwargs)

Bases: django.db.models.base.Model

Form wizard form entry.

A coupling point between *FormWizardEntry* and *FormEntry*.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

form_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

form_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

form_wizard_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via ForwardOneToOneDescriptor subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

Child.parent is a ForwardManyToOneDescriptor instance.

form_wizard_entry_id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

position

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

class fobi.models.**FormWizardHandler**(*args, **kwargs)

Bases: [fobi.models.AbstractPluginModel](#)

Form wizard handler plugin. Used when `fobi.settings.RESTRICT_PLUGIN_ACCESS` is set to `True`.

Properties

- *plugin_uid* (str): Plugin UID.
- *users* (django.contrib.auth.models.User): White list of the users allowed to use the form handler plugin.
- *groups* (django.contrib.auth.models.Group): White list of the user groups allowed to use the form handler plugin.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

get_registered_plugins()

Add choices.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = <django.db.models.manager.Manager object>

plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`Pizza.toppings` and `Topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class `fobi.models.ModelFormWizardHandlerEntry(*args, **kwargs)`

Bases: `fobi.models.AbstractFormWizardPluginEntry`

Form wizard handler entry.

Properties

- `form_wizard_entry` (`fobi.models.ModelFormWizardEntry`): FormWizard to which the handler plugin belongs to.
- `plugin_uid` (str): Plugin UID.
- `plugin_data` (str): JSON formatted string with plugin data.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_wizard_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one (via `ForwardOneToOneDescriptor` subclass) relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`Child.parent` is a `ForwardManyToOneDescriptor` instance.

get_registered_plugins()

Gets registered plugins.

get_registry()

Get registry.

id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

objects = `<django.db.models.manager.Manager object>`

plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

48.4.21 fobi.settings module

- `RESTRICT_PLUGIN_ACCESS` (bool): If set to True, (Django) permission system for fobi plugins is enabled.
- `FORM_ELEMENT_PLUGINS_MODULE_NAME` (str): Name of the module to placed in the (external) apps in which the fobi form element plugin code should be implemented and registered.
- `FORM_HANDLER_PLUGINS_MODULE_NAME` (str): Name of the module to placed in the (external) apps in which the fobi form handler plugin code should be implemented and registered.
- `FORM_CALLBACKS_MODULE_NAME` (str): Name of the module to placed in the (external) apps in which the fobi form callback code should be implemented and registered.

- *FORM_HANDLER_PLUGINS_EXECUTION_ORDER* (tuple): Order in which the form handler plugins are to be executed.
- *FORM_WIZARD_HANDLER_PLUGINS_EXECUTION_ORDER* (tuple): Order in which the form handler plugins are to be executed.
- *DEBUG*

48.4.22 fobi.test module

48.4.23 fobi.utils module

Another helper module. This module can NOT be safely imported from any fobi (sub)module - thus should be imported carefully.

`fobi.utils.append_edit_and_delete_links_to_field(form_element_plugin, form_element_entry, origin=None, extra={}, widget_cls=None)`

Append edit and delete links to the field.

Should return dictionary, which would be used to update default kwargs of form fields.

The hidden inputs *form-{counter}-position* and *form-{counter}-id* are for saving the ordering of the elements (*position* field).

Return dict

`fobi.utils.get_allowed_form_element_plugin_uids(user)`

Get allowed form element plugin uids.

`fobi.utils.get_allowed_form_handler_plugin_uids(user)`

Get allowed form handler plugin uids.

`fobi.utils.get_allowed_form_wizard_handler_plugin_uids(user)`

Get allowed form wizard handler plugin uids.

`fobi.utils.get_allowed_plugin_uids(plugin_model_cls, user)`

Get allowed plugins uids for user given.

Parameters

- **plugin_model_cls** (`fobi.models.AbstractPluginModel`) – Subclass of `fobi.models.AbstractPluginModel`.
- **user** (`django.contrib.auth.models.User`) –

Return list

`fobi.utils.get_user_form_element_plugin_uids(user)`

Get user form element plugin uids.

`fobi.utils.get_user_form_element_plugins(user)`

Get user form element plugins.

`fobi.utils.get_user_form_element_plugins_grouped(user, sort_by_value=False)`

Get user form element plugins grouped.

`fobi.utils.get_user_form_field_plugin_uids(user)`

Get user form element plugin uids.

`fobi.utils.get_user_form_handler_plugin_uids(user)`

Get user form handler plugin uids.

`fobi.utils.get_user_form_handler_plugins(user, exclude_used_singles=False, used_form_handler_plugin_uids=[])`

Get list of plugins allowed for user.

Parameters

- **user** (*django.contrib.auth.models.User*) –
- **exclude_used_singles** (*bool*) –
- **used_form_handler_plugin_uids** (*list*) –

Return list

`fobi.utils.get_user_form_handler_plugins_grouped(user, sort_by_value=False)`

Get user form handler plugins grouped.

`fobi.utils.get_user_form_wizard_handler_plugin_uids(user)`

Get user form handler plugin uids.

`fobi.utils.get_user_form_wizard_handler_plugins(user, exclude_used_singles=False, used_form_wizard_handler_plugin_uids=[])`

Get list of plugins allowed for user.

Parameters

- **user** (*django.contrib.auth.models.User*) –
- **exclude_used_singles** (*bool*) –
- **used_form_wizard_handler_plugin_uids** (*list*) –

Return list

`fobi.utils.get_user_form_wizard_handler_plugins_grouped(user)`

Get user form wizard handler plugins grouped.

`fobi.utils.get_user_plugin_uids(get_allowed_plugin_uids_func, get_registered_plugin_uids_func, registry, user)`

Gets a list of user plugin uids as a list.

If not yet auto-discovered, auto-discovers them.

Parameters

- **get_allowed_plugin_uids_func** (*callable*) –
- **get_registered_plugin_uids_func** (*callable*) –
- **registry** (*fobi.base.BaseRegistry*) – Subclass of *fobi.base.BaseRegistry* instance.
- **user** (*django.contrib.auth.models.User*) –

Return list

`fobi.utils.get_user_plugins(get_allowed_plugin_uids_func, get_registered_plugins_func, registry, user)`

Get user plugins.

Gets a list of user plugins in a form if tuple (plugin name, plugin description). If not yet autodiscovered, autodiscovered them.

Parameters

- **get_allowed_plugin_uids_func** (*callable*) –
- **get_registered_plugins_func** (*callable*) –

- **registry** (`fobi.base.BaseRegistry`) – Subclass of `fobi.base.BaseRegistry` instance.
- **user** (`django.contrib.auth.models.User`) –

Return list

`fobi.utils.get_user_plugins_grouped`(`get_allowed_plugin_uids_func`, `get_registered_plugins_grouped_func`, `registry`, `user`, `sort_items=True`, `sort_by_value=False`)

Get user plugins grouped.

Parameters

- **get_allowed_plugin_uids_func** (`callable`) –
- **get_registered_plugins_grouped_func** (`callable`) –
- **registry** (`fobi.base.BaseRegistry`) – Subclass of `fobi.base.BaseRegistry` instance.
- **user** (`django.contrib.auth.models.User`) –
- **sort_items** (`bool`) –
- **sort_by_value** (`bool`) –

Return dict

`fobi.utils.get_wizard_files_upload_dir`()

Get absolute path to the upload directory of fobi form wizard files.

If `WIZARD_FILES_UPLOAD_DIR` path is absolute, return as is. Otherwise, prepend `BASE_PATH`.

Return str Absolute path.

`fobi.utils.perform_form_entry_import`(`request`, `form_data`)

Perform form entry import.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_data** (`dict`) –

:return :class:`fobi.modes.FormEntry`: Instance of.

`fobi.utils.prepare_form_entry_export_data`(`form_entry`, `form_element_entries=None`, `form_handler_entries=None`)

Prepare form entry export data.

Parameters

- **form_entry** (`fobi.modes.FormEntry`) – Instance of.
- **form_element_entries** (`django.db.models.QuerySet`) – QuerySet of `FormElementEntry` instances.
- **form_handler_entries** (`django.db.models.QuerySet`) – QuerySet of `FormHandlerEntry` instances.

Return str

`fobi.utils.sync_plugins`()

Sync registered plugins.

Syncs the registered plugin list with data in `fobi.models.FormFieldPluginModel`, `fobi.models.FormHandlerPluginModel` and `fobi.models.FormWizardHandlerPluginModel`.

48.4.24 fobi.validators module

`fobi.validators.url_exists(url, local=False)`
Check if URL exists.

Parameters

- `url` (*str*) –
- `local` (*bool*) –

Return `bool`

48.4.25 fobi.views module

48.4.26 fobi.widgets module

`class fobi.widgets.BooleanRadioSelect(*args, **kwargs)`
Bases: `django.forms.widgets.RadioSelect`
Boolean radio select for Django.

Example

```
>>> class DummyForm(forms.Form):
>>>     agree = forms.BooleanField(label=_("Agree?"),
>>>                               required=False,
>>>                               widget=BooleanRadioSelect)
```

property `media`

`class fobi.widgets.NumberInput(attrs=None)`
Bases: `django.forms.widgets.Input`
`input_type = 'number'`

property `media`

`template_name = 'django/forms/widgets/number.html'`

`class fobi.widgets.RichSelect(attrs=None, choices=(), prepend_html=None, append_html=None, override_name=None)`
Bases: `django.forms.widgets.Select`

Rich select widget with some rich enhancements.

Based on original `Select` widget and intended to be a drop-off replacement.

property `media`

`render(name, value, attrs=None, **kwargs)`
Renders the element, having prepended and appended extra parts.

`class fobi.widgets.RichSelectInverseQuotes(attrs=None, choices=(), prepend_html=None, append_html=None, override_name=None)`

Bases: `fobi.widgets.RichSelect`

Almost same as original, but uses alternative `flatatt` function.

Uses inverse quotes.

property `media`

```
option_template_name = 'fobi/django/forms/widgets/rich_select_inverse_option.html'  
template_name = 'fobi/django/forms/widgets/rich_select_inverse.html'
```

48.4.27 Module contents

INDICES AND TABLES

- [genindex](#)
- [modindex](#)
- [search](#)

BIBLIOGRAPHY

- [1.1] Dashboard
- [1.2] Create a form
- [1.3] Edit form - form elements tab active, no elements yet
- [1.4] Edit form - form elements tab active, add a form element menu
- [1.5] Edit form - add a form element (URL plugin)
- [1.6] Edit form - form elements tab active, with form elements
- [1.7] Edit form - form handlers tab active, no handlers yet
- [1.8] Edit form - form handlers tab active, add form handler menu
- [1.9] Edit form - add a form handler (Mail plugin)
- [1.10] Edit form - form handlers tab active, with form handlers
- [1.11] Edit form - form properties tab active
- [1.12] View form
- [1.13] View form - form submitted (thanks page)
- [1.14] Edit form - add a form element (Video plugin)
- [1.15] Edit form - add a form element (Boolean plugin)
- [1.16] Edit form
- [1.17] View form
- [2.1] Edit form - form elements tab active, with form elements
- [2.2] Edit form - form elements tab active, add a form element menu
- [2.3] Edit form - add a form element (Hidden plugin)
- [2.4] Edit form - form handlers tab active, with form handlers
- [2.5] Edit form - form properties tab active
- [2.6] View form

PYTHON MODULE INDEX

f

fobi, 555
fobi.admin, 493
fobi.app, 498
fobi.apps, 498
fobi.base, 498
fobi.conf, 517
fobi.constants, 518
fobi.context_processors, 518
fobi.contrib, 472
fobi.contrib.apps, 327
fobi.contrib.apps.djangocms_integration, 272
fobi.contrib.apps.djangocms_integration.apps, 271
fobi.contrib.apps.djangocms_integration.conf, 271
fobi.contrib.apps.djangocms_integration.defaults, 272
fobi.contrib.apps.djangocms_integration.helpers, 272
fobi.contrib.apps.djangocms_integration.settings, 272
fobi.contrib.apps.drf_integration, 323
fobi.contrib.apps.drf_integration.apps, 317
fobi.contrib.apps.drf_integration.base, 317
fobi.contrib.apps.drf_integration.dynamic, 319
fobi.contrib.apps.drf_integration.fields, 319
fobi.contrib.apps.drf_integration.form_elements, 314
fobi.contrib.apps.drf_integration.form_elements.Content, 279
fobi.contrib.apps.drf_integration.form_elements.Content_Image, 273
fobi.contrib.apps.drf_integration.form_elements.Content_Image.apps, 272
fobi.contrib.apps.drf_integration.form_elements.Content_Image.base, 273
fobi.contrib.apps.drf_integration.form_elements.Content_Image.fobi_integration_form_elements, 273
fobi.contrib.apps.drf_integration.form_elements.Content_Image_urls, 274
fobi.contrib.apps.drf_integration.form_elements.content, 273
fobi.contrib.apps.drf_integration.form_elements.content, 274
fobi.contrib.apps.drf_integration.form_elements.content, 274
fobi.contrib.apps.drf_integration.form_elements.content, 276
fobi.contrib.apps.drf_integration.form_elements.content, 274
fobi.contrib.apps.drf_integration.form_elements.content, 275
fobi.contrib.apps.drf_integration.form_elements.content, 275
fobi.contrib.apps.drf_integration.form_elements.content, 277
fobi.contrib.apps.drf_integration.form_elements.content, 276
fobi.contrib.apps.drf_integration.form_elements.content, 276
fobi.contrib.apps.drf_integration.form_elements.content, 276
fobi.contrib.apps.drf_integration.form_elements.content, 278
fobi.contrib.apps.drf_integration.form_elements.content, 277
fobi.contrib.apps.drf_integration.form_elements.content, 277
fobi.contrib.apps.drf_integration.form_elements.content, 277
fobi.contrib.apps.drf_integration.form_elements.content, 279
fobi.contrib.apps.drf_integration.form_elements.content, 278
fobi.contrib.apps.drf_integration.form_elements.content, 278
fobi.contrib.apps.drf_integration.form_elements.content, 278
fobi.contrib.apps.drf_integration.form_elements.fields, 314
fobi.contrib.apps.drf_integration.form_elements.fields, 280

[illegible]

[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 322
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 311
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 311
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 311
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 313
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 312
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 312
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 312
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 314
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 313
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 313
[fobi.contrib.apps.drf_integration.form_elements.fields.rich_textarea](#), 313
[fobi.contrib.apps.drf_integration.form_handlers](#), 317
[fobi.contrib.apps.drf_integration.form_handlers.db_store](#), 315
[fobi.contrib.apps.drf_integration.form_handlers.db_store](#), 314
[fobi.contrib.apps.drf_integration.form_handlers.db_store](#), 314
[fobi.contrib.apps.drf_integration.form_handlers.http_post](#), 316
[fobi.contrib.apps.drf_integration.form_handlers.http_post](#), 315
[fobi.contrib.apps.drf_integration.form_handlers.http_post](#), 315
[fobi.contrib.apps.drf_integration.form_handlers.http_post](#), 315
[fobi.contrib.apps.drf_integration.form_handlers.mail](#), 317
[fobi.contrib.apps.drf_integration.form_handlers.mail](#), 316
[fobi.contrib.apps.drf_integration.form_handlers.mail](#), 316
[fobi.contrib.apps.drf_integration.form_handlers.mail](#), 316
[fobi.contrib.apps.drf_integration.helpers](#), 321
[fobi.contrib.apps.drf_integration.metadata](#), 321
[fobi.contrib.apps.drf_integration.serializers](#), 321

[fobi.contrib.apps.drf_integration.urls](#), 322
[fobi.contrib.apps.drf_integration.utils](#), 322
[fobi.contrib.apps.drf_integration.views](#), 322
[fobi.contrib.apps.feincms_integration](#), 325
[fobi.contrib.apps.feincms_integration.form_elements](#), 323
[fobi.contrib.apps.feincms_integration.conf](#), 323
[fobi.contrib.apps.feincms_integration.defaults](#), 323
[fobi.contrib.apps.feincms_integration.helpers](#), 323
[fobi.contrib.apps.feincms_integration.form_elements](#), 324
[fobi.contrib.apps.feincms_integration.widgets](#), 324
[fobi.contrib.apps.mezzanine_integration](#), 327
[fobi.contrib.apps.mezzanine_integration.apps](#), 326
[fobi.contrib.apps.mezzanine_integration.conf](#), 326
[fobi.contrib.apps.mezzanine_integration.defaults](#), 326
[fobi.contrib.apps.mezzanine_integration.helpers](#), 326
[fobi.contrib.apps.mezzanine_integration.settings](#), 326
[fobi.contrib.plugins](#), 451
[fobi.contrib.plugins.form_elements](#), 428
[fobi.contrib.plugins.form_elements.content](#), 428
[fobi.contrib.plugins.form_elements.content.content_image](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.a](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.b](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.c](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.d](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.e](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.f](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.g](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.h](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.i](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.j](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.k](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.l](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.m](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.n](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.o](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.p](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.q](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.r](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.s](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.t](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.u](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.v](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.w](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.x](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.y](#), 428
[fobi.contrib.plugins.form_elements.content.content_image.z](#), 428

331	342
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.content.content_video.b
331	343
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.content.content_video.c
332	343
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.content.content_video.c
332	344
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.content.content_video.f
332	344
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.content.content_video.f
333	344
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.content.content_video.s
336	345
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields,
333	420
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.boolean,
334	347
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.boolean.apps,
335	345
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.boolean.base,
336	345
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.boolean.fobi_for
336	346
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.boolean.forms,
340	346
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
336	350
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
337	347
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
338	347
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
339	348
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
339	348
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
342	348
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
340	349
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.checkbox_select,
340	350
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.date,
341	352
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.date.apps,
341	350
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.date.base,
341	350
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.date.fobi_form_e
342	351
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.date.forms,
342	351
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.date.widgets,
345	352
fobi.contrib.plugins.form_elements.content.conf	fobi.contrib.plugins.form_elements.fields.date_drop_down,

355	363
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.file.base,
352	364
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.file.conf,
352	364
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.file.defaults,
353	365
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.file.fobi_form_e
354	365
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.file.forms,
357	366
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.file.settings,
355	367
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.float,
355	369
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.float.apps,
355	367
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.float.base,
356	367
fobi.contrib.plugins.form_elements.fields.date	fobi.contrib.plugins.form_elements.fields.float.fobi_form
357	367
fobi.contrib.plugins.form_elements.fields.decimal	fobi.contrib.plugins.form_elements.fields.float.forms,
359	368
fobi.contrib.plugins.form_elements.fields.decimal	fobi.contrib.plugins.form_elements.fields.hidden,
357	370
fobi.contrib.plugins.form_elements.fields.decimal	fobi.contrib.plugins.form_elements.fields.hidden.apps,
357	369
fobi.contrib.plugins.form_elements.fields.decimal	fobi.contrib.plugins.form_elements.fields.hidden.base,
358	369
fobi.contrib.plugins.form_elements.fields.decimal	fobi.contrib.plugins.form_elements.fields.hidden.fobi_form
358	369
fobi.contrib.plugins.form_elements.fields.duration	fobi.contrib.plugins.form_elements.fields.hidden.forms,
362	370
fobi.contrib.plugins.form_elements.fields.duration	fobi.contrib.plugins.form_elements.fields.input,
359	373
fobi.contrib.plugins.form_elements.fields.duration	fobi.contrib.plugins.form_elements.fields.input.apps,
359	370
fobi.contrib.plugins.form_elements.fields.duration	fobi.contrib.plugins.form_elements.fields.input.base,
360	371
fobi.contrib.plugins.form_elements.fields.duration	fobi.contrib.plugins.form_elements.fields.input.constants,
361	371
fobi.contrib.plugins.form_elements.fields.email	fobi.contrib.plugins.form_elements.fields.input.fobi_form
363	371
fobi.contrib.plugins.form_elements.fields.email	fobi.contrib.plugins.form_elements.fields.input.forms,
362	372
fobi.contrib.plugins.form_elements.fields.email	fobi.contrib.plugins.form_elements.fields.integer,
362	375
fobi.contrib.plugins.form_elements.fields.email	fobi.contrib.plugins.form_elements.fields.integer.apps,
362	373
fobi.contrib.plugins.form_elements.fields.email	fobi.contrib.plugins.form_elements.fields.integer.base,
363	373
fobi.contrib.plugins.form_elements.fields.file	fobi.contrib.plugins.form_elements.fields.integer.fobi_for
367	374
fobi.contrib.plugins.form_elements.fields.file	fobi.contrib.plugins.form_elements.fields.integer.forms,

412 fobi.contrib.plugins.form_elements.fields.slug_forms, 421
 412 fobi.contrib.plugins.form_elements.fields.text_forms, 425
 415 fobi.contrib.plugins.form_elements.fields.text_forms_apps, 422
 413 fobi.contrib.plugins.form_elements.fields.text_forms_base, 422
 413 fobi.contrib.plugins.form_elements.fields.text_forms_base_apps, 423
 414 fobi.contrib.plugins.form_elements.fields.text_forms_base_plugins, 423
 414 fobi.contrib.plugins.form_elements.fields.text_forms_forms, 423
 416 fobi.contrib.plugins.form_elements.fields.text_forms_fobi, 423
 415 fobi.contrib.plugins.form_elements.fields.text_forms_apps, 424
 415 fobi.contrib.plugins.form_elements.fields.text_forms_base, 424
 415 fobi.contrib.plugins.form_elements.fields.text_forms_base_plugins, 426
 416 fobi.contrib.plugins.form_elements.fields.text_forms_forms, 425
 418 fobi.contrib.plugins.form_elements.fields.time_forms, 425
 416 fobi.contrib.plugins.form_elements.fields.time_forms_apps, 425
 417 fobi.contrib.plugins.form_elements.fields.time_forms_base, 426
 417 fobi.contrib.plugins.form_elements.fields.time_forms_base_plugins, 428
 418 fobi.contrib.plugins.form_elements.fields.time_forms_forms, 428
 418 fobi.contrib.plugins.form_elements.fields.url, 426
 420 fobi.contrib.plugins.form_elements.fields.url_apps, 427
 418 fobi.contrib.plugins.form_elements.fields.url_base, 427
 419 fobi.contrib.plugins.form_elements.fields.url_fobi_form_elements, 427
 419 fobi.contrib.plugins.form_elements.fields.url_fobi_form_elements_widgets, 427
 419 fobi.contrib.plugins.form_elements.fields.url_fobi_form_elements_handlers, 447
 419 fobi.contrib.plugins.form_elements.fields.url_fobi_form_elements_handlers_db_store, 438
 426 fobi.contrib.plugins.form_elements.security, fobi.contrib.plugins.form_handlers.db_store.admin, 429
 422 fobi.contrib.plugins.form_elements.security.captcha, fobi.contrib.plugins.form_handlers.db_store.apps, 430
 420 fobi.contrib.plugins.form_elements.security.captcha_apps, fobi.contrib.plugins.form_handlers.db_store.base, 430
 420 fobi.contrib.plugins.form_elements.security.captcha_base, fobi.contrib.plugins.form_handlers.db_store.callbacks, 432
 421 fobi.contrib.plugins.form_elements.security.captcha_base_plugins, fobi.contrib.plugins.form_handlers.db_store.conf, 432
 421 fobi.contrib.plugins.form_elements.security.captcha_forms, fobi.contrib.plugins.form_handlers.db_store.defaults, 432

454 fobi.contrib.themes.foundation5.widgets.form_elements.date
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements,
455 fobi.contrib.themes.foundation5.widgets.form_elements.date
fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget,
456 fobi.contrib.themes.foundation5.widgets.form_elements.dumm
fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps,
455 fobi.contrib.themes.foundation5.widgets.form_elements.dumm
fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_form_elements,
455 fobi.contrib.themes.foundation5.widgets.form_elements.dumm
fobi.contrib.themes.djangocms_admin_style_theme, 465
462 fobi.contrib.themes.foundation5.widgets.form_handlers,
fobi.contrib.themes.djangocms_admin_style_theme.apps, 466
460 fobi.contrib.themes.foundation5.widgets.form_handlers.db_s
fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes,
460 fobi.contrib.themes.foundation5.widgets.form_handlers.db_s
fobi.contrib.themes.djangocms_admin_style_theme.widgets, 455
460 fobi.contrib.themes.foundation5.widgets.form_handlers.db_s
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements,
459 fobi.contrib.themes.simple, 472
fobi.contrib.themes.djangocms_admin_style_theme.widgets.richtext_admin_style_widg
459 fobi.contrib.themes.simple.fobi_themes, 470
fobi.contrib.themes.djangocms_admin_style_theme.widgets.richtext_admin_style_widg
458 fobi.contrib.themes.simple.widgets.form_elements,
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_style_widg
459 fobi.contrib.themes.simple.widgets.form_elements.content_r
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers,
460 fobi.contrib.themes.simple.widgets.form_elements.content_r
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store,
460 fobi.contrib.themes.simple.widgets.form_elements.content_r
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps,
459 fobi.contrib.themes.simple.widgets.form_handlers,
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements,
459 fobi.contrib.themes.simple.widgets.form_handlers.db_store,
fobi.contrib.themes.foundation5, 468 470
fobi.contrib.themes.foundation5.apps, 466 fobi.contrib.themes.simple.widgets.form_handlers.db_store.
fobi.contrib.themes.foundation5.fobi_themes, 469
466 fobi.contrib.themes.simple.widgets.form_handlers.db_store.
fobi.contrib.themes.foundation5.widgets, 466 469
fobi.contrib.themes.foundation5.widgets.form_elements.data_structures, 518
465 fobi.decorators, 520
fobi.contrib.themes.foundation5.widgets.form_elements.richtext_foundation5_widget,
463 fobi.discover, 521
fobi.contrib.themes.foundation5.widgets.form_elements.richtext_foundation5_widget.apps,
462 fobi.exceptions, 522
fobi.contrib.themes.foundation5.widgets.form_elements.richtext_foundation5_widget.fobi_form_ele
463 fobi.form_utils, 524
fobi.contrib.themes.foundation5.widgets.form_elements.foundation5_widget,
464 fobi.helpers, 530
fobi.contrib.themes.foundation5.widgets.form_elements.foundation5_widget.apps,
463 fobi.integration.helpers, 472
fobi.contrib.themes.foundation5.widgets.form_elements.foundation5_widget.fobi_form_elements,
463 fobi.management, 474
fobi.contrib.themes.foundation5.widgets.form_elements.management.commands.fobi_find_broken_entries,
464 fobi.management.commands.fobi_find_broken_entries,

- 473
- fobi.management.commands.fobi_sync_plugins, 474
- fobi.management.commands.fobi_update_plugin_data, 474
- fobi.migrations, 479
- fobi.migrations.0001_initial, 474
- fobi.migrations.0002_auto_20150912_1744, 476
- fobi.migrations.0003_auto_20160517_1005, 476
- fobi.migrations.0004_auto_20160906_1513, 476
- fobi.migrations.0005_auto_20160908_1457, 477
- fobi.migrations.0006_auto_20160911_1549, 477
- fobi.migrations.0007_auto_20160926_1652, 477
- fobi.migrations.0008_formwizardhandlerentry, 478
- fobi.migrations.0009_formwizardentry_wizard_type, 478
- fobi.migrations.0010_formwizardhandler, 478
- fobi.migrations.0011_formentry_title, 478
- fobi.migrations.0012_auto_20161109_1550, 479
- fobi.migrations.0013_formwizardentry_show_all_navigation_buttons, 479
- fobi.migrations.0014_auto_20170516_1413, 479
- fobi.models, 536
- fobi.reusable, 480
- fobi.reusable.markdown_widget, 480
- fobi.reusable.markdown_widget.helpers, 479
- fobi.reusable.markdown_widget.widgets, 480
- fobi.settings, 550
- fobi.templatetags, 482
- fobi.templatetags.fobi_tags, 480
- fobi.templatetags.future_compat, 482
- fobi.tests, 487
- fobi.tests.base, 482
- fobi.tests.constants, 483
- fobi.tests.core, 483
- fobi.tests.data, 483
- fobi.tests.helpers, 483
- fobi.tests.test_browser_build_dynamic_forms, 484
- fobi.tests.test_core, 485
- fobi.tests.test_drf_integration, 485
- fobi.tests.test_dynamic_forms, 486
- fobi.tests.test_feincms_integration, 486
- fobi.tests.test_form_importers_mailchimp, 486
- fobi.tests.test_sortable_dict, 487
- fobi.urls, 487
- fobi.urls.edit, 487
- fobi.urls.view, 487
- fobi.utils, 551
- fobi.validators, 554
- fobi.views, 554
- fobi.widgets, 554
- fobi.wizard, 493
- fobi.wizard.views, 493
- fobi.wizard.views.dynamic, 487
- fobi.wizard.views.views, 492

INDEX

A

abstract (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget.Meta attribute), 324
 abstract (fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormEntry.Meta attribute), 434
 abstract (fobi.models.AbstractFormWizardPluginEntry.Meta attribute), 536
 abstract (fobi.models.AbstractPluginEntry.Meta attribute), 536
 abstract (fobi.models.AbstractPluginModel.Meta attribute), 537
 abstract (fobi.models.BaseAbstractPluginEntry.Meta attribute), 538
 AbstractFormWizardPluginEntry (class in fobi.models), 536
 AbstractFormWizardPluginEntry.Meta (class in fobi.models), 536
 AbstractPluginEntry (class in fobi.models), 536
 AbstractPluginEntry.Meta (class in fobi.models), 536
 AbstractPluginModel (class in fobi.models), 537
 AbstractPluginModel.Meta (class in fobi.models), 537
 AbstractSavedFormEntry (class in fobi.contrib.plugins.form_handlers.db_store.models), 434
 AbstractSavedFormEntry.Meta (class in fobi.contrib.plugins.form_handlers.db_store.models), 434
 action (fobi.models.FormEntry attribute), 541
 actions (fobi.admin.FormElementAdmin attribute), 493
 actions (fobi.admin.FormHandlerAdmin attribute), 495
 actions (fobi.admin.FormWizardHandlerAdmin attribute), 497
 actions (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormEntryAdmin attribute), 429
 actions (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntryAdmin attribute), 430
 active_date_from (fobi.models.FormEntry attribute), 541
 active_date_to (fobi.models.FormEntry attribute), 541
 add_arguments() (fobi.management.commands.fobi_sync_plugins.Command method), 474
 add_form_element_entry_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 456
 add_form_element_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.fobi_themes attribute), 460
 add_form_element_entry_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 466
 add_form_element_entry_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 470
 add_form_element_entry_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 456
 add_form_element_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.fobi_themes attribute), 460
 add_form_element_entry_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 466
 add_form_element_entry_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 470
 add_form_handler_entry_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 456
 add_form_handler_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.fobi_themes attribute), 460
 add_form_handler_entry_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 466
 add_form_handler_entry_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 470
 add_form_handler_entry_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 456
 add_form_handler_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.fobi_themes attribute), 460

[illegible]

[illegible]

BaseContentRichTextPluginWidget (class in [fobi.admin](#)), 497
[fobi.contrib.plugins.form_elements.content.content_rich_text.plugin_widgets](#), 339
 bulk_change_form_wizard_handler_plugins() (in module [fobi.admin](#)), 497
 BaseDataStorage (class in [fobi.base](#)), 498
 BaseDatePluginWidget (class in [fobi.admin.BasePluginModelAdmin](#) method), [fobi.contrib.plugins.form_elements.fields.date.widgets](#), 493
 BaseDateTimePluginWidget (class in [fobi.forms](#)), 524
[fobi.contrib.plugins.form_elements.fields.datetime.widgets](#), 357
 BulkChangeFormElementPluginsForm (class in [fobi.forms](#)), 524
 BulkChangeFormElementPluginsForm.Meta (class in [fobi.forms](#)), 524
 BaseDbStorePluginWidget (class in [BulkChangeFormHandlerPluginsForm](#) (class in [fobi.forms](#)), 525
[fobi.contrib.plugins.form_handlers.db_store.widgets](#)), 438
 BulkChangeFormHandlerPluginsForm.Meta (class in [fobi.forms](#)), 525
 BaseDummyPluginWidget (class in [BulkChangeFormWizardHandlerPluginsForm](#) (class in [fobi.forms](#)), 525
[fobi.contrib.plugins.form_elements.test.dummy.widgets](#)), 427
 BulkChangeFormWizardHandlerPluginsForm.Meta (class in [fobi.forms](#)), 525
 BaseException, 522
 BaseFobiBrowserBuldDynamicFormsTest (class in [fobi.tests.base](#)), 482
 BaseFormFieldPluginForm (class in [fobi.base](#)), 498
 BaseFormImporter (class in [fobi.form_importers](#)), 523
 basename ([fobi.contrib.apps.drf_integration.views.FobiFormEntryView](#) attribute), 322
 BasePlugin (class in [fobi.base](#)), 499
 BasePluginForm (class in [fobi.base](#)), 502
 BasePluginModelAdmin (class in [fobi.admin](#)), 493
 BasePluginModelAdmin.Meta (class in [fobi.admin](#)), 493
 BaseRegistry (class in [fobi.base](#)), 503
 BaseSliderPluginWidget (class in [fobi.contrib.plugins.form_elements.fields.slider.widgets](#)), 411
 BooleanRadioSelect (class in [fobi.widgets](#)), 554
 BooleanSelectForm (class in [fobi.contrib.plugins.form_elements.fields.boolean.forms](#)), 346
 BooleanSelectPlugin (class in [fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base](#)), 279
 BooleanSelectPlugin (class in [fobi.contrib.plugins.form_elements.fields.boolean.base](#)), 280
 BooleanSelectPlugin (class in [fobi.contrib.plugins.form_elements.fields.boolean.base](#)), 345
 BooleanSelectPlugin (class in [fobi.contrib.plugins.form_elements.fields.boolean.base](#)), 346
 Bootstrap3Theme (class in [fobi.contrib.themes.bootstrap3.fobi_themes](#)), 456
 bulk_change_form_element_plugins() (in module [fobi.admin](#)), 497
 bulk_change_form_handler_plugins() (in module

C
 callback() ([fobi.base.FormCallback](#) method), 504
 callback() ([fobi.base.IntegrationFormCallback](#) method), 509
 callback() ([fobi.contrib.plugins.form_handlers.db_store.callbacks.AutoForm](#) method), 432
 callback() ([fobi.contrib.plugins.form_handlers.mail.callbacks.AutoForm](#) method), 443
 can_redirect ([fobi.contrib.apps.feincms_integration.widgets.FobiFormW](#) attribute), 324
 can_redirect ([fobi.integration.processors.IntegrationProcessor](#) attribute), 472
 CaptchaInputForm (class in [fobi.contrib.plugins.form_elements.security.captcha.forms](#)), 421
 CaptchaInputPlugin (class in [fobi.contrib.plugins.form_elements.security.captcha.base](#)), 421
 CaptchaInputPlugin (class in [fobi.contrib.plugins.form_elements.security.captcha.fobi_form_el](#)), 421
 CaptchaInputPlugin (class in [fobi.contrib.plugins.form_elements.security.captcha.fobi_form_el](#)), 421
 CheckboxSelectMultipleInputForm (class in [fobi.contrib.plugins.form_elements.fields.checkbox_select_multipl](#)), 349
 CheckboxSelectMultipleInputPlugin (class in [fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_s](#)), 280
 CheckboxSelectMultipleInputPlugin (class in [fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_s](#)), 281
 CheckboxSelectMultipleInputPlugin (class in [fobi.contrib.plugins.form_elements.fields.checkbox_select_multipl](#)), 347
 CheckboxSelectMultipleInputPlugin (class in [fobi.contrib.plugins.form_elements.fields.checkbox_select_multipl](#))

- 348
- `ClassProperty` (class in `fobi.base`), 504
- `classproperty` (in module `fobi.base`), 511
- `clean()` (`fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm` method), 358
- `clean()` (`fobi.contrib.plugins.form_elements.fields.duration.forms.DurationInputForm` method), 361
- `clean()` (`fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm` method), 363
- `clean()` (`fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm` method), 366
- `clean()` (`fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm` method), 368
- `clean()` (`fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm` method), 370
- `clean()` (`fobi.contrib.plugins.form_elements.fields.input.forms.InputForm` method), 372
- `clean()` (`fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm` method), 374
- `clean()` (`fobi.contrib.plugins.form_elements.fields.ip_address.forms.IPAddressInputForm` method), 376
- `clean()` (`fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm` method), 380
- `clean()` (`fobi.contrib.plugins.form_elements.fields.range.forms.RangeInputForm` method), 385
- `clean()` (`fobi.contrib.plugins.form_elements.fields.regex.forms.RegexForm` method), 387
- `clean()` (`fobi.contrib.plugins.form_elements.fields.slider.forms.SliderForm` method), 410
- `clean()` (`fobi.contrib.plugins.form_elements.fields.slug.forms.SlugForm` method), 413
- `clean()` (`fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm` method), 414
- `clean()` (`fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm` method), 420
- `clean()` (`fobi.contrib.plugins.form_elements.security.honeycomb.forms.HoneycombForm` method), 423
- `clean_action()` (`fobi.forms.FormEntryForm` method), 527
- `clean_dict()` (in module `fobi.helpers`), 531
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.checkbox.forms.CheckboxInputForm` method), 349
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm` method), 351
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm` method), 356
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.radio.forms.RadioInputForm` method), 383
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm` method), 390
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleInputForm` method), 398
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms.SelectMultipleWithMaxInputForm` method), 407
- `clean_initial()` (`fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm` method), 418
- `clean_text()` (`fobi.contrib.plugins.form_elements.content.content_richtext.forms.RichTextForm` method), 349
- `clean_text()` (`fobi.contrib.plugins.form_elements.content.content_text.forms.TextForm` method), 349
- `cleans_up_after_itself` (`fobi.contrib.plugins.form_elements.content.content_image.base.ContentImageBaseForm` attribute), 482
- `clone_file()` (in module `fobi.helpers`), 532
- `clone_plugin_data()` (`fobi.base.BasePlugin` method), 499
- `clone_plugin_data()` (`fobi.contrib.plugins.form_elements.content.content_image.base.ContentImageBaseForm` method), 327
- `clone_plugin_data()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_firebase_plugin.fobi_firebase_plugin` method), 328
- `code_of_conduct()` (`fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField` attribute), 444
- `collect_plugin_data()` (in module `fobi.base`), 511
- `combine_dicts()` (in module `fobi.helpers`), 532
- `commands` (`fobi.management.commands.fobi_find_broken_entries`), 473
- `commands` (`fobi.management.commands.fobi_sync_plugins`), 474
- `commands` (`fobi.management.commands.fobi_update_plugin_data`), 474
- `config_list()` (`fobi.wizard.views.dynamic.DynamicWizardView` method), 489
- `condition_dict` (`fobi.wizard.views.dynamic.DynamicWizardView` attribute), 489
- `Config` (class in `fobi.apps`), 498
- `Config` (class in `fobi.contrib.apps.django_cms_integration.apps`), 271
- `Config` (class in `fobi.contrib.apps.drf_integration.apps`), 317
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.forms.RichTextForm`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.content.content_text.forms.TextForm`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.content.content_image.base.ContentImageBaseForm`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.content.content_image.fobi_firebase_plugin.fobi_firebase_plugin`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.boolean.forms.BooleanField`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.boolean_with_max.forms.BooleanFieldWithMax`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.checkbox.forms.CheckboxForm`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.date.forms.DateField`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.datetime.forms.DateTimeField`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.radio.forms.RadioForm`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select.forms.SelectForm`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.forms.SelectMultipleForm`), 273
- `Config` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.forms.SelectMultipleWithMaxForm`), 273

280	312
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.url.apps),	Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.url.apps),
282	313
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.db_store.apps),	Config (class in fobi.contrib.apps.drf_integration.form_handlers.db_store.apps),
283	314
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.http_repo.apps),	Config (class in fobi.contrib.apps.drf_integration.form_handlers.http_repo.apps),
284	315
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.mail.apps),	Config (class in fobi.contrib.apps.drf_integration.form_handlers.mail.apps),
285	316
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.feincms_integration.apps),	Config (class in fobi.contrib.apps.feincms_integration.apps),
286	323
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.mezzanine_integration.apps),	Config (class in fobi.contrib.apps.mezzanine_integration.apps),
287	325
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.content_image.apps),	Config (class in fobi.contrib.plugins.form_elements.content.content_image.apps),
288	327
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.content_image.apps),	Config (class in fobi.contrib.plugins.form_elements.content.content_image.apps),
290	330
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.content_markdown.apps),	Config (class in fobi.contrib.plugins.form_elements.content.content_markdown.apps),
291	333
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.content_richtext.apps),	Config (class in fobi.contrib.plugins.form_elements.content.content_richtext.apps),
292	336
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.content_text.apps),	Config (class in fobi.contrib.plugins.form_elements.content.content_text.apps),
293	340
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.content_video.apps),	Config (class in fobi.contrib.plugins.form_elements.content.content_video.apps),
294	342
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.boolean.apps),
295	345
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.checkbox_select.apps),
296	347
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.date.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.date.apps),
297	350
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.date_drop_down.apps),
298	352
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.datetime.apps),
299	355
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.decimal.apps),
300	357
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.duration.apps),
301	359
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.email.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.email.apps),
303	362
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.file.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.file.apps),
304	363
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.float.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.float.apps),
306	367
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.hidden.apps),
308	369
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.input.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.input.apps),
309	370
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.integer.apps),
310	373
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.ip_address.apps),
311	375
Config (class in fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps),	Config (class in fobi.contrib.plugins.form_elements.fields.null_boolean.apps),

377	Config (class in fobi.contrib.plugins.form_elements.fields.password.apps), 451
379	Config (class in fobi.contrib.plugins.form_elements.fields.radio.apps), 452
381	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 453
383	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 453
386	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 454
387	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 455
390	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 460
393	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 458
395	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 459
399	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 466
401	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 462
404	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 463
408	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 464
411	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 464
413	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 465
415	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 470
416	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 468
418	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 469
420	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 319
422	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 273
425	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 327
426	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 328
430	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 327
438	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 328
441	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 327
447	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 328
456	Config (class in fobi.contrib.plugins.form_elements.fields.range.apps), 328
	ContentImageField (class in fobi.contrib.plugins.form_elements.content.content_image.fields), 319
	ContentImageForm (class in fobi.contrib.plugins.form_elements.content.content_image.forms), 273
	ContentImagePlugin (class in fobi.contrib.plugins.form_elements.content.content_image.plugin), 327
	ContentImageURLForm (class in fobi.contrib.plugins.form_elements.content.content_image_url.form), 328

ContentImageURLPlugin	(class in ContentRichTextPluginWidget	(class in ContentRichTextPluginWidget
fobi.contrib.apps.drf_integration.form_elements.content.content_image_url_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url_plugin)
ContentImageURLPlugin	(class in ContentRichTextPluginWidget	(class in ContentRichTextPluginWidget
fobi.contrib.apps.drf_integration.form_elements.content.content_image_url_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_image_url_plugin)
ContentImageURLPlugin	(class in ContentRichTextPluginWidget	(class in ContentRichTextPluginWidget
fobi.contrib.plugins.form_elements.content.content_image_url_plugin)	fobi.contrib.plugins.form_elements.content.content_image_url_plugin)	fobi.contrib.plugins.form_elements.content.content_image_url_plugin)
ContentImageURLPlugin	(class in ContentTextField	(class in ContentTextField
fobi.contrib.plugins.form_elements.content.content_image_url_plugin)	fobi.contrib.plugins.form_elements.content.content_image_url_plugin)	fobi.contrib.plugins.form_elements.content.content_image_url_plugin)
ContentMarkdownField	(class in ContentTextForm	(class in ContentTextForm
fobi.contrib.apps.drf_integration.fields),	fobi.contrib.plugins.form_elements.content.content_text.forms),	fobi.contrib.plugins.form_elements.content.content_text.forms),
ContentMarkdownForm	(class in ContentTextPlugin	(class in ContentTextPlugin
fobi.contrib.plugins.form_elements.content.content_markdown_form)	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_form)	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_form)
ContentMarkdownPlugin	(class in ContentTextPlugin	(class in ContentTextPlugin
fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_plugin)
ContentMarkdownPlugin	(class in ContentTextPlugin	(class in ContentTextPlugin
fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_plugin)	fobi.contrib.apps.drf_integration.form_elements.content.content_markdown_plugin)
ContentMarkdownPlugin	(class in ContentTextPlugin	(class in ContentTextPlugin
fobi.contrib.plugins.form_elements.content.content_markdown_plugin)	fobi.contrib.plugins.form_elements.content.content_text.fobi_form)	fobi.contrib.plugins.form_elements.content.content_text.fobi_form)
ContentMarkdownPlugin	(class in ContentVideoField	(class in ContentVideoField
fobi.contrib.plugins.form_elements.content.content_markdown_plugin)	fobi.contrib.plugins.form_elements.content.content_markdown_plugin)	fobi.contrib.plugins.form_elements.content.content_markdown_plugin)
ContentMarkdownPluginWidget	(class in ContentVideoForm	(class in ContentVideoForm
fobi.contrib.themes.bootstrap3.widgets.form_elements.content.content_markdown_plugin_widget)	fobi.contrib.plugins.form_elements.content.content_markdown_plugin_widget)	fobi.contrib.plugins.form_elements.content.content_markdown_plugin_widget)
ContentRichTextField	(class in ContentVideoPlugin	(class in ContentVideoPlugin
fobi.contrib.apps.drf_integration.fields),	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext_field)	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext_field)
ContentRichTextForm	(class in ContentVideoPlugin	(class in ContentVideoPlugin
fobi.contrib.plugins.form_elements.content.content_richtext_form)	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext_form)	fobi.contrib.apps.drf_integration.form_elements.content.content_richtext_form)
ContentRichTextPlugin	(class in ContentVideoPlugin	(class in ContentVideoPlugin
fobi.contrib.apps.drf_integration.form_elements.content.content_richtext_plugin)	fobi.contrib.plugins.form_elements.content.content_video.base),	fobi.contrib.plugins.form_elements.content.content_video.base),
ContentRichTextPlugin	(class in ContentVideoPlugin	(class in ContentVideoPlugin
fobi.contrib.apps.drf_integration.form_elements.content.content_richtext_plugin)	fobi.contrib.plugins.form_elements.content.content_video.fobi_form)	fobi.contrib.plugins.form_elements.content.content_video.fobi_form)
ContentRichTextPlugin	(class in convert_to_markdown()	(in module
fobi.contrib.plugins.form_elements.content.content_richtext_plugin)	fobi.contrib.plugins.form_elements.content.content_richtext_plugin)	fobi.contrib.plugins.form_elements.content.content_richtext_plugin)
ContentRichTextPlugin	(class in CookieWizardView	(class in fobi.wizard.views.views),
fobi.contrib.plugins.form_elements.content.content_richtext_plugin)	fobi_form_elements),	fobi_form_elements),
ContentRichTextPluginWidget	(class in create_form_entry_ajax_template	(class in create_form_entry_ajax_template
fobi.contrib.themes.bootstrap3.widgets.form_elements.content.content_richtext_plugin_widget)	fobi.contrib.themes.bootstrap3.widgets.form_elements.content.content_richtext_plugin_widget)	fobi.contrib.themes.bootstrap3.widgets.form_elements.content.content_richtext_plugin_widget)

`fobi.contrib.plugins.form_elements.fields.datetime.datetime` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 356

`DateTimeInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 285

`DateTimeInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 355

`DateTimeInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 357

`DateTimeInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 355

`DateTimePluginWidget` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 358

`DateTimePluginWidget` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 454

`DateTimePluginWidget` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 464

`DateTimeSelectPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 336

`DateTimeSelectPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 284

`DateTimeSelectPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 284

`db_clean_up()` (in module `fobi.tests.helpers`), 483

`DBStoreHandlerPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 314

`DBStoreHandlerPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 314

`DBStoreHandlerPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 314

`DBStoreHandlerPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 430

`DBStoreHandlerPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 432

`DbStorePluginWidget` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 459

`DbStorePluginWidget` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 466

`DbStorePluginWidget` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 469

`DBStoreWizardHandlerPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 431

`DBStoreWizardHandlerPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 433

`DecimalInputForm` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 358

`DecimalInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.datetime.datetime`), 285

[illegible]

[attribute](#)), 477
[dependencies](#) ([fobi.migrations.0008_formwizardhandlerentry.Migration](#) [attribute](#)), 478
[dependencies](#) ([fobi.migrations.0009_formwizardentry_wizard_type.Migration](#) [attribute](#)), 478
[dependencies](#) ([fobi.migrations.0010_formwizardhandler.Migration](#) [attribute](#)), 478
[dependencies](#) ([fobi.migrations.0011_formentry_title.Migration](#) [attribute](#)), 478
[dependencies](#) ([fobi.migrations.0012_auto_20161109_1550.Migration](#) [attribute](#)), 479
[dependencies](#) ([fobi.migrations.0013_formwizardentry_shadow.Migration](#) [attribute](#)), 479
[dependencies](#) ([fobi.migrations.0014_auto_20170516_1413.Migration](#) [attribute](#)), 479
[description](#) ([fobi.base.BasePlugin](#) [attribute](#)), 499
[description](#) ([fobi.contrib.apps.drf_integration.views.FobiFormEntryView](#) [attribute](#)), 322
[description](#) ([fobi.form_importers.BaseFormImporter](#) [attribute](#)), 523
[detail](#) ([fobi.contrib.apps.drf_integration.views.FobiFormEntryView](#) [attribute](#)), 322
[dispatch\(\)](#) ([fobi.wizard.views.dynamic.DynamicWizardView](#) [method](#)), 489
[DjangoCMSAdminStyleTheme](#) (class in [fobi.contrib.themes.djangocms_admin_style_theme.FobiAdminStyleTheme](#)), 460
[do_http_repost\(\)](#) ([fobi.contrib.plugins.form_handlers.http_repost.HTTPRepostHandlerPlugin](#) [method](#)), 438
[do_http_repost\(\)](#) ([fobi.contrib.plugins.form_handlers.http_repost.HTTPRepostHandlerPlugin](#) [method](#)), 440
[do_slugify\(\)](#) (in module [fobi.helpers](#)), 532
[DoesNotExist](#), 522
[done\(\)](#) ([fobi.contrib.plugins.form_importers.mailchimp_importer.views.MailchimpImporterWizardView](#) [method](#)), 450
[done\(\)](#) ([fobi.wizard.views.dynamic.DynamicWizardView](#) [method](#)), 489
[done_step_name](#) ([fobi.wizard.views.dynamic.DynamicNamedUrlWizardView](#) [attribute](#)), 488
[DRFIntegrationFormElementPluginProcessor](#) (class in [fobi.contrib.apps.drf_integration.base](#)), 317
[DRFSubmitPluginFormDataMixin](#) (class in [fobi.contrib.apps.drf_integration.base](#)), 317
[DummyPlugin](#) (class in [fobi.contrib.plugins.form_elements.test.dummy.base](#)), 427
[DummyPlugin](#) (class in [fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements](#)), 427
[DummyPluginWidget](#) (class in [fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements](#)), 455
[DummyPluginWidget](#) (class in [fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_elements](#)), 455
[fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_elements](#), 455
[DurationInputForm](#) (class in [fobi.contrib.plugins.form_elements.fields.duration.forms](#)), 361
[DurationInputPlugin](#) (class in [fobi.contrib.apps.drf_integration.form_elements.fields.duration.base](#)), 286
[DurationInputPlugin](#) (class in [fobi.contrib.plugins.form_elements.fields.duration.base](#)), 287
[DurationInputPlugin](#) (class in [fobi.contrib.plugins.form_elements.fields.duration.base](#)), 287
[DurationInputPlugin](#) (class in [fobi.contrib.plugins.form_elements.fields.duration.base](#)), 287
[DynamicCookieWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 487
[DynamicNamedUrlCookieWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 487
[DynamicNamedUrlSessionWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 487
[DynamicNamedUrlWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 488
[DynamicSessionWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 488
[DynamicWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 488
[DynamicWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 488

E

[edit_form_element_entry_ajax_template](#) (class in [fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#) [attribute](#)), 456
[edit_form_element_entry_ajax_template](#) (class in [fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.FobiAdminStyleTheme](#) [attribute](#)), 460
[edit_form_element_entry_ajax_template](#) (class in [fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#) [attribute](#)), 467
[edit_form_element_entry_ajax_template](#) (class in [fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#) [attribute](#)), 470
[edit_form_element_entry_template](#) (class in [fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#) [attribute](#)), 456
[edit_form_element_entry_template](#) (class in [fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.FobiAdminStyleTheme](#) [attribute](#)), 460
[edit_form_element_entry_template](#) (class in [fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#) [attribute](#)), 467

<code>edit_form_element_entry_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> attribute), 470	<code>edit_form_handler_entry_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> attribute), 470
<code>edit_form_entry_ajax_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 456	<code>edit_form_template</code> (<code>fobi.base.BasePlugin</code> attribute), 499
<code>edit_form_entry_ajax_template</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme</code> attribute), 460	<code>edit_form_wizard_entry_ajax_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 457
<code>edit_form_entry_ajax_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> attribute), 467	<code>edit_form_wizard_entry_ajax_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> attribute), 467
<code>edit_form_entry_ajax_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> attribute), 470	<code>edit_form_wizard_entry_ajax_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> attribute), 470
<code>edit_form_entry_edit_option_html()</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme</code> class method), 460	<code>edit_form_wizard_entry_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 457
<code>edit_form_entry_help_text_extra()</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme</code> class method), 461	<code>edit_form_wizard_entry_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> attribute), 467
<code>edit_form_entry_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 456	<code>edit_form_wizard_entry_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> attribute), 471
<code>edit_form_entry_template</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme</code> attribute), 461	<code>edit_form_wizard_handler_entry_ajax_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 457
<code>edit_form_entry_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> attribute), 467	<code>edit_form_wizard_handler_entry_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 457
<code>edit_form_entry_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> attribute), 470	<code>email</code> (<code>fobi.helpers.StrippedUser</code> property), 531
<code>edit_form_handler_entry_ajax_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 456	<code>EmailInputForm</code> (class in <code>fobi.contrib.plugins.form_elements.fields.email.forms</code>), 363
<code>edit_form_handler_entry_ajax_template</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme</code> attribute), 461	<code>EmailInputPlugin</code> (class in <code>fobi.contrib.apps.drf_integration.form_elements.fields.email.base</code>), 288
<code>edit_form_handler_entry_ajax_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> attribute), 467	<code>EmailInputPlugin</code> (class in <code>fobi.contrib.apps.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme</code>), 288
<code>edit_form_handler_entry_ajax_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> attribute), 470	<code>EmailInputPlugin</code> (class in <code>fobi.contrib.plugins.form_elements.fields.email.base</code>), 362
<code>edit_form_handler_entry_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 457	<code>embed_form_entry_submitted_ajax_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> attribute), 457
<code>edit_form_handler_entry_template</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme</code> attribute), 461	<code>ensure_autodiscover()</code> (in module <code>fobi.base</code>), 511
<code>edit_form_handler_entry_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> attribute), 467	<code>ensure_autodiscover()</code> (in module <code>fobi.base</code>), 511
	<code>ensure_unique_filename()</code> (in module <code>fobi.helpers</code>), 524
	<code>entry_user</code> (<code>fobi.models.AbstractFormWizardPluginEntry</code>), 524

property), 536	fail_on_missing_plugin (fobi.base.BaseRegistry attribute), 503
entry_user (fobi.models.AbstractPluginEntry property), 536	fail_on_missing_plugin (fobi.base.FormElementPluginRegistry attribute), 506
entry_user (fobi.models.BaseAbstractPluginEntry property), 538	fail_on_missing_plugin (fobi.base.FormHandlerPluginRegistry attribute), 507
ErrorDict (class in fobi.form_utils), 524	fail_on_missing_plugin (fobi.base.FormWizardHandlerPluginRegistry attribute), 509
ErrorList (class in fobi.form_utils), 524	fail_on_missing_plugin (fobi.base.IntegrationFormElementPluginRegistry attribute), 510
export() (fobi.helpers.JSONDataExporter method), 530	fail_on_missing_plugin (fobi.base.IntegrationFormHandlerPluginRegistry attribute), 511
export_entries_icon_class (fobi.contrib.plugins.form_handlers.db_store.widgets.DbStorePluginWidget attribute), 438	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_entries_icon_class (fobi.contrib.themes.djangocms_admin_style_theme.widgets.DbStorePluginWidget attribute), 459	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_entries_icon_class (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store.widgets.DbStorePluginWidget attribute), 466	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_entries_icon_class (fobi.contrib.themes.simple.widgets.form_handlers.db_store.widgets.DbStorePluginWidget attribute), 469	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_saved_form_data_entries() (in module fobi.contrib.plugins.form_handlers.db_store.views), 437	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_saved_form_wizard_data_entries() (in module fobi.contrib.plugins.form_handlers.db_store.views), 437	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_to_csv() (fobi.contrib.plugins.form_handlers.db_store.widgets.DbStorePluginWidget method), 434	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_to_json() (fobi.helpers.JSONDataExporter method), 530	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
export_to_xls() (fobi.contrib.plugins.form_handlers.db_store.widgets.DbStorePluginWidget method), 434	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extra (fobi.admin.FormElementEntryInlineAdmin attribute), 494	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extra (fobi.admin.FormHandlerEntryInlineAdmin attribute), 496	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extra (fobi.admin.FormWizardFormEntryInlineAdmin attribute), 497	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extra (fobi.admin.FormWizardHandlerEntryInlineAdmin attribute), 497	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extract_field_properties() (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter method), 448	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extract_field_properties() (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter method), 448	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extract_field_properties() (fobi.form_importers.BaseFormImporter method), 523	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
extract_file_path() (in module fobi.helpers), 532	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
F	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448
fail_on_missing_plugin (fobi.base.BaseRegistry attribute), 503	fail_on_missing_plugin (fobi.contrib.plugins.db_store.fobi_form_elements.DbStorePluginWidget attribute), 448

fields (*fobi.forms.ModelFormEntryForm.Meta* attribute), 527

fields (*fobi.forms.FormHandlerEntryForm.Meta* attribute), 527

fields (*fobi.forms.FormHandlerForm.Meta* attribute), 528

fields (*fobi.forms.FormWizardEntryForm.Meta* attribute), 528

fields (*fobi.forms.FormWizardFormEntryForm.Meta* attribute), 529

fields (*fobi.forms.FormWizardHandlerEntryForm.Meta* attribute), 529

fields_mapping (*fobi.contrib.plugins.form_importers.mailchimp_importer* attribute), 448

fields_mapping (*fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter* attribute), 448

fields_mapping (*fobi.form_importers.BaseFormImporter* attribute), 523

fieldsets (*fobi.admin.BasePluginModelAdmin* attribute), 493

fieldsets (*fobi.admin.FormElementEntryAdmin* attribute), 494

fieldsets (*fobi.admin.FormEntryAdmin* attribute), 494

fieldsets (*fobi.admin.FormFieldsetEntryAdmin* attribute), 495

fieldsets (*fobi.admin.FormHandlerEntryAdmin* attribute), 496

fieldsets (*fobi.admin.FormWizardEntryAdmin* attribute), 496

fieldsets (*fobi.contrib.plugins.form_handlers.db_store.admin.SocialFormEntryAdmin* attribute), 429

fieldsets (*fobi.contrib.plugins.form_handlers.db_store.admin.SocialFormWizardEntryAdmin* attribute), 430

FileInputForm (class in *fobi.contrib.plugins.form_elements.fields.file.forms*), 366

FileInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.file*), 289

FileInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.file.base*), 289

FileInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.file.fobi_form_importers*), 365

filter_horizontal (*fobi.admin.BasePluginModelAdmin* attribute), 493

finalize() (*fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget* method), 324

fire_form_callbacks() (in module *fobi.base*), 511

firstof() (in module *fobi.templatetags.future_compat*),

flatatt_inverse_quotes() (in module *fobi.helpers*), 532

FloatInputForm (class in *fobi.contrib.plugins.form_elements.fields.float.forms*), 368

FloatInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.float.base*), 290

FloatInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_form_importers*), 291

FloatInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.float.fobi_form_element*), 367

fobi module, 555

fobi.admin module, 493

fobi.app module, 498

fobi.apps module, 498

fobi.base module, 498

fobi.conf module, 498

fobi.constants module, 518

fobi.contrib module, 472

fobi.contrib.apps module, 327

fobi.contrib.apps.djangocms_integration module, 271

fobi.contrib.apps.djangocms_integration.conf module, 271

fobi.contrib.apps.djangocms_integration.defaults module, 272

fobi.contrib.apps.djangocms_integration.helpers module, 272

fobi.contrib.apps.djangocms_integration.settings module, 272

fobi.contrib.apps.drf_integration module, 323

fobi.contrib.apps.drf_integration.apps module, 317

fobi.contrib.apps.drif_integration.form_elements.fobidist.durationapps	fbi.edst.durapps.drif_integration.form_elements.fields.int
module, 287	module, 294
fobi.contrib.apps.drif_integration.form_elements.fobidist.durationapps	fbi.edst.durapps.drif_integration.form_elements.fields.ip
module, 286	module, 295
fobi.contrib.apps.drif_integration.form_elements.fobidist.durationbase	fbi.edst.durapps.baseintegration.form_elements.fields.ip
module, 286	module, 294
fobi.contrib.apps.drif_integration.form_elements.fobidist.durationapps	fbi.integrationofformelementsfields.ip
module, 287	module, 294
fobi.contrib.apps.drif_integration.form_elements.fobidist.eiaapps	fbi.edst.eiaapps.drif_integration.form_elements.fields.ip
module, 288	module, 295
fobi.contrib.apps.drif_integration.form_elements.fobidist.eiaappsdrf	fbi.edst.eiaappsdrf_integration.form_elements.fields.null
module, 287	module, 296
fobi.contrib.apps.drif_integration.form_elements.fobidist.eiaappsbase	fbi.edst.eiaappsbase_drif_integration.form_elements.fields.null
module, 288	module, 295
fobi.contrib.apps.drif_integration.form_elements.fobidist.eiaappsdrf	fbi.edst.eiaappsdrf_integrationofformelements.fields.null
module, 288	module, 295
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillapps	fbi.edst.fillapps.drif_integration.form_elements.fields.null
module, 290	module, 296
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillappsdrf	fbi.edst.fillappsdrf_integration.form_elements.fields.pas
module, 288	module, 297
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillappsbase	fbi.edst.fillappsbase_drif_integration.form_elements.fields.pas
module, 289	module, 296
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillappsfobidistintegrationofformelements	fbi.edst.fillappsfobidistintegrationofformelements.fields.pas
module, 289	module, 296
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillapps	fbi.edst.fillapps.drif_integration.form_elements.fields.pas
module, 291	module, 297
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillappsdrf	fbi.edst.fillappsdrf_integration.form_elements.fields.rac
module, 290	module, 298
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillappsbase	fbi.edst.fillappsbase_drif_integration.form_elements.fields.rac
module, 290	module, 297
fobi.contrib.apps.drif_integration.form_elements.fobidist.fillappsfobidistintegrationofformelements	fbi.edst.fillappsfobidistintegrationofformelements.fields.rac
module, 291	module, 297
fobi.contrib.apps.drif_integration.form_elements.fobidist.hildapps	fbi.edst.hildapps.drif_integration.form_elements.fields.rac
module, 292	module, 298
fobi.contrib.apps.drif_integration.form_elements.fobidist.hildappsdrf	fbi.edst.hildappsdrf_integration.form_elements.fields.rar
module, 291	module, 299
fobi.contrib.apps.drif_integration.form_elements.fobidist.hildappsbase	fbi.edst.hildappsbase_drif_integration.form_elements.fields.rar
module, 291	module, 298
fobi.contrib.apps.drif_integration.form_elements.fobidist.hildappsfobidistintegrationofformelements	fbi.edst.hildappsfobidistintegrationofformelements.fields.rar
module, 292	module, 298
fobi.contrib.apps.drif_integration.form_elements.fobidist.inlapps	fbi.edst.inlapps.drif_integration.form_elements.fields.rar
module, 293	module, 299
fobi.contrib.apps.drif_integration.form_elements.fobidist.inlappsdrf	fbi.edst.inlappsdrf_integration.form_elements.fields.reg
module, 292	module, 300
fobi.contrib.apps.drif_integration.form_elements.fobidist.inlappsbase	fbi.edst.inlappsbase_drif_integration.form_elements.fields.reg
module, 292	module, 299
fobi.contrib.apps.drif_integration.form_elements.fobidist.inlappsfobidistintegrationofformelements	fbi.edst.inlappsfobidistintegrationofformelements.fields.reg
module, 293	module, 299
fobi.contrib.apps.drif_integration.form_elements.fobidist.inlapps	fbi.edst.inlapps.drif_integration.form_elements.fields.reg
module, 294	module, 300
fobi.contrib.apps.drif_integration.form_elements.fobidist.inlappsdrf	fbi.edst.inlappsdrf_integration.form_elements.fields.sel
module, 293	module, 301
fobi.contrib.apps.drif_integration.form_elements.fobidist.inlappsbase	fbi.edst.inlappsbase_drif_integration.form_elements.fields.sel
module, 293	module, 300

fobi.contrib.apps.drf_integration.form_handlers module, 316
 fobi.contrib.apps.drf_integration.helpers module, 321
 fobi.contrib.apps.drf_integration.metadata module, 321
 fobi.contrib.apps.drf_integration.serializers module, 321
 fobi.contrib.apps.drf_integration.urls module, 322
 fobi.contrib.apps.drf_integration.utils module, 322
 fobi.contrib.apps.drf_integration.views module, 322
 fobi.contrib.apps.feincms_integration module, 325
 fobi.contrib.apps.feincms_integration.apps module, 323
 fobi.contrib.apps.feincms_integration.conf module, 323
 fobi.contrib.apps.feincms_integration.defaults module, 323
 fobi.contrib.apps.feincms_integration.helpers module, 323
 fobi.contrib.apps.feincms_integration.settings module, 324
 fobi.contrib.apps.feincms_integration.widgets module, 324
 fobi.contrib.apps.mezzanine_integration module, 327
 fobi.contrib.apps.mezzanine_integration.apps module, 325
 fobi.contrib.apps.mezzanine_integration.conf module, 326
 fobi.contrib.apps.mezzanine_integration.defaults module, 326
 fobi.contrib.apps.mezzanine_integration.helpers module, 326
 fobi.contrib.apps.mezzanine_integration.settings module, 326
 fobi.contrib.plugins module, 451
 fobi.contrib.plugins.form_elements module, 428
 fobi.contrib.plugins.form_elements.content module, 345
 fobi.contrib.plugins.form_elements.content.conf module, 330
 fobi.contrib.plugins.form_elements.content.conf.apps module, 327
 fobi.contrib.plugins.form_elements.content.conf.base module, 327
 fobi.contrib.plugins.form_elements.content.conf.conf module, 328
 fobi.contrib.plugins.form_elements.content.conf.handlers module, 328
 fobi.contrib.plugins.form_elements.content.conf.helpers module, 328
 fobi.contrib.plugins.form_elements.content.conf.metadata module, 329
 fobi.contrib.plugins.form_elements.content.conf.serializers module, 330
 fobi.contrib.plugins.form_elements.content.conf.urls module, 330
 fobi.contrib.plugins.form_elements.content.conf.utils module, 333
 fobi.contrib.plugins.form_elements.content.conf.views module, 330
 fobi.contrib.plugins.form_elements.content.conf.widgets module, 331
 fobi.contrib.plugins.form_elements.content.conf.apps module, 331
 fobi.contrib.plugins.form_elements.content.conf.conf module, 332
 fobi.contrib.plugins.form_elements.content.conf.defaults module, 332
 fobi.contrib.plugins.form_elements.content.conf.helpers module, 332
 fobi.contrib.plugins.form_elements.content.conf.settings module, 333
 fobi.contrib.plugins.form_elements.content.conf.widgets module, 336
 fobi.contrib.plugins.form_elements.content.conf.markdown module, 333
 fobi.contrib.plugins.form_elements.content.conf.markdown.apps module, 334
 fobi.contrib.plugins.form_elements.content.conf.markdown.conf module, 335
 fobi.contrib.plugins.form_elements.content.conf.markdown.defaults module, 336
 fobi.contrib.plugins.form_elements.content.conf.markdown.helpers module, 336
 fobi.contrib.plugins.form_elements.content.conf.markdown.settings module, 340
 fobi.contrib.plugins.form_elements.content.conf.richtext module, 336
 fobi.contrib.plugins.form_elements.content.conf.richtext.apps module, 337
 fobi.contrib.plugins.form_elements.content.conf.richtext.conf module, 338
 fobi.contrib.plugins.form_elements.content.conf.richtext.conf.apps module, 339
 fobi.contrib.plugins.form_elements.content.conf.richtext.conf.base module, 339
 fobi.contrib.plugins.form_elements.content.conf.richtext.conf.conf module, 342
 fobi.contrib.plugins.form_elements.content.conf.richtext.conf.handlers module, 340

fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.checkbox_select
 module, 340 module, 350
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date
 module, 341 module, 352
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date.apps
 module, 341 module, 350
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date.base
 module, 341 module, 350
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date.fobi_form_e
 module, 342 module, 351
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date.forms
 module, 342 module, 351
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date.widgets
 module, 345 module, 352
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date_drop_down
 module, 342 module, 355
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date_drop_down.a
 module, 343 module, 352
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date_drop_down.b
 module, 343 module, 352
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date_drop_down.f
 module, 344 module, 353
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.date_drop_down.f
 module, 344 module, 354
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.datetime
 module, 344 module, 357
 fobi.contrib.plugins.form_elements.content.conf_fobi.contrib.plugins.form_elements.fields.datetime.apps
 module, 345 module, 355
 fobi.contrib.plugins.form_elements.fields fobi.contrib.plugins.form_elements.fields.datetime.base
 module, 420 module, 355
 fobi.contrib.plugins.form_elements.fields.bool_fobi.contrib.plugins.form_elements.fields.datetime.fobi_for
 module, 347 module, 355
 fobi.contrib.plugins.form_elements.fields.bool_fobi.contrib.plugins.form_elements.fields.datetime.forms
 module, 345 module, 356
 fobi.contrib.plugins.form_elements.fields.bool_fobi.contrib.plugins.form_elements.fields.datetime.widgets
 module, 345 module, 357
 fobi.contrib.plugins.form_elements.fields.bool_fobi.contrib.plugins.form_elements.fields.decimal
 module, 346 module, 359
 fobi.contrib.plugins.form_elements.fields.bool_fobi.contrib.plugins.form_elements.fields.decimal.apps
 module, 346 module, 357
 fobi.contrib.plugins.form_elements.fields.checkbox_fobi.contrib.plugins.form_elements.fields.decimal.base
 module, 350 module, 357
 fobi.contrib.plugins.form_elements.fields.checkbox_fobi.contrib.plugins.form_elements.fields.decimal.fobi_for
 module, 347 module, 358
 fobi.contrib.plugins.form_elements.fields.checkbox_fobi.contrib.plugins.form_elements.fields.decimal.forms
 module, 347 module, 358
 fobi.contrib.plugins.form_elements.fields.checkbox_fobi.contrib.plugins.form_elements.fields.duration
 module, 348 module, 362
 fobi.contrib.plugins.form_elements.fields.checkbox_fobi.contrib.plugins.form_elements.fields.duration.apps
 module, 348 module, 359
 fobi.contrib.plugins.form_elements.fields.checkbox_fobi.contrib.plugins.form_elements.fields.duration.base
 module, 348 module, 359
 fobi.contrib.plugins.form_elements.fields.checkbox_fobi.contrib.plugins.form_elements.fields.duration.fobi_for
 module, 349 module, 360

fobi.contrib.plugins.form_elements.fields.duration module, 361

fobi.contrib.plugins.form_elements.fields.email module, 363

fobi.contrib.plugins.form_elements.fields.emailapps module, 362

fobi.contrib.plugins.form_elements.fields.emailbase module, 362

fobi.contrib.plugins.form_elements.fields.emailfobi module, 362

fobi.contrib.plugins.form_elements.fields.emailforms module, 363

fobi.contrib.plugins.form_elements.fields.file module, 367

fobi.contrib.plugins.form_elements.fields.fileapps module, 363

fobi.contrib.plugins.form_elements.fields.filebase module, 364

fobi.contrib.plugins.form_elements.fields.fileconf module, 364

fobi.contrib.plugins.form_elements.fields.filedefaults module, 365

fobi.contrib.plugins.form_elements.fields.filefobicontrib module, 365

fobi.contrib.plugins.form_elements.fields.fileforms module, 366

fobi.contrib.plugins.form_elements.fields.filesettings module, 367

fobi.contrib.plugins.form_elements.fields.float module, 369

fobi.contrib.plugins.form_elements.fields.floatapps module, 367

fobi.contrib.plugins.form_elements.fields.floatbase module, 367

fobi.contrib.plugins.form_elements.fields.floatfobicontrib module, 367

fobi.contrib.plugins.form_elements.fields.floatforms module, 368

fobi.contrib.plugins.form_elements.fields.hidden module, 370

fobi.contrib.plugins.form_elements.fields.hiddenapps module, 369

fobi.contrib.plugins.form_elements.fields.hiddenbase module, 369

fobi.contrib.plugins.form_elements.fields.hiddenfobicontrib module, 369

fobi.contrib.plugins.form_elements.fields.hiddenforms module, 369

fobi.contrib.plugins.form_elements.fields.hiddenfobicontrib module, 370

fobi.contrib.plugins.form_elements.fields.input module, 373

fobi.contrib.plugins.form_elements.fields.inputapps module, 370

fobi.contrib.plugins.form_elements.fields.inputbase module, 370

fobi.contrib.plugins.form_elements.fields.inputfobicontrib module, 371

fobi.contrib.plugins.form_elements.fields.inputforms module, 371

fobi.contrib.plugins.form_elements.fields.inputfobicontrib module, 371

fobi.contrib.plugins.form_elements.fields.inputforms module, 372

fobi.contrib.plugins.form_elements.fields.integer module, 375

fobi.contrib.plugins.form_elements.fields.integerapps module, 373

fobi.contrib.plugins.form_elements.fields.integerbase module, 373

fobi.contrib.plugins.form_elements.fields.integerfobi module, 374

fobi.contrib.plugins.form_elements.fields.integerforms module, 374

fobi.contrib.plugins.form_elements.fields.ip_address module, 377

fobi.contrib.plugins.form_elements.fields.ip_addressapps module, 375

fobi.contrib.plugins.form_elements.fields.ip_addressbase module, 375

fobi.contrib.plugins.form_elements.fields.ip_addressfobi module, 375

fobi.contrib.plugins.form_elements.fields.ip_addressforms module, 376

fobi.contrib.plugins.form_elements.fields.null_boolean module, 379

fobi.contrib.plugins.form_elements.fields.null_booleanapps module, 377

fobi.contrib.plugins.form_elements.fields.null_booleanbase module, 377

fobi.contrib.plugins.form_elements.fields.null_booleanfobicontrib module, 378

fobi.contrib.plugins.form_elements.fields.null_booleanforms module, 381

fobi.contrib.plugins.form_elements.fields.password module, 379

fobi.contrib.plugins.form_elements.fields.passwordapps module, 379

fobi.contrib.plugins.form_elements.fields.passwordbase module, 379

fobi.contrib.plugins.form_elements.fields.passwordfobicontrib module, 380

fobi.contrib.plugins.form_elements.fields.passwordforms module, 380

fobi.contrib.plugins.form_elements.fields.radio module, 383

fobi.contrib.plugins.form_elements.fields.radioapps module, 381

fobi.contrib.plugins.form_elements.fields.radiobase module, 381

fobi.contrib.plugins.form_elements.fields.radioconf module, 381


```

fobi.contrib.plugins.form_elements.fields.url.fobi.contrib.plugins.form_elements.test.dummy.widgets
    module, 419
fobi.contrib.plugins.form_elements.fields.url.fobicontrib.plugins.form_handlers
    module, 419
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store
    module, 426
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.admin
    module, 422
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.apps
    module, 420
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.base
    module, 421
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.callbacks
    module, 421
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.conf
    module, 421
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.defaults
    module, 425
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.fobi_form_han
    module, 422
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.helpers
    module, 422
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.migrations
    module, 423
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.migrations.000
    module, 423
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.migrations.000
    module, 423
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.models
    module, 423
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.settings
    module, 424
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.urls
    module, 424
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.urls.form_han
    module, 426
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.urls.form_wiza
    module, 425
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.views
    module, 425
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.form_handlers.db_store.widgets
    module, 425
fobi.contrib.plugins.form_elements.security.fobicontrib.plugins.http_repost
    module, 426
fobi.contrib.plugins.form_elements.test.fobicontrib.plugins.form_handlers.http_repost.apps
    module, 428
fobi.contrib.plugins.form_elements.test.dummy.fobicontrib.plugins.form_handlers.http_repost.base
    module, 428
fobi.contrib.plugins.form_elements.test.dummy.fobicontrib.plugins.form_handlers.http_repost.fobi_form_h
    module, 426
fobi.contrib.plugins.form_elements.test.dummy.fobicontrib.plugins.form_handlers.http_repost.forms
    module, 427
fobi.contrib.plugins.form_elements.test.dummy.fobicontrib.plugins.form_handlers.mail
    module, 427

```

fobi.contrib.plugins.form_handlers.mail.apps	fobi.contrib.themes.bootstrap3.widgets.form_elements.conte
module, 441	module, 452
fobi.contrib.plugins.form_handlers.mail.base	fobi.contrib.themes.bootstrap3.widgets.form_elements.conte
module, 442	module, 453
fobi.contrib.plugins.form_handlers.mail.callbacks	fobi.contrib.themes.bootstrap3.widgets.form_elements.conte
module, 443	module, 452
fobi.contrib.plugins.form_handlers.mail.conf	fobi.contrib.themes.bootstrap3.widgets.form_elements.conte
module, 443	module, 452
fobi.contrib.plugins.form_handlers.mail.defaults	fobi.contrib.themes.bootstrap3.widgets.form_elements.date
module, 444	module, 453
fobi.contrib.plugins.form_handlers.mail.fields	fobi.contrib.themes.bootstrap3.widgets.form_elements.date
module, 444	module, 453
fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers	fobi.contrib.themes.bootstrap3.widgets.form_elements.date
module, 444	module, 453
fobi.contrib.plugins.form_handlers.mail.forms	fobi.contrib.themes.bootstrap3.widgets.form_elements.datet
module, 445	module, 454
fobi.contrib.plugins.form_handlers.mail.helpers	fobi.contrib.themes.bootstrap3.widgets.form_elements.datet
module, 446	module, 453
fobi.contrib.plugins.form_handlers.mail.mixins	fobi.contrib.themes.bootstrap3.widgets.form_elements.datet
module, 446	module, 454
fobi.contrib.plugins.form_handlers.mail.settings	fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy
module, 447	module, 455
fobi.contrib.plugins.form_handlers.mail.widgets	fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy
module, 447	module, 454
fobi.contrib.plugins.form_importers	fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy
module, 451	module, 455
fobi.contrib.plugins.form_importers.mailchimp_importer	fobi.contrib.themes.bootstrap3.widgets.form_elements.slide
module, 451	module, 456
fobi.contrib.plugins.form_importers.mailchimp_importer_app	fobi.contrib.themes.bootstrap3.widgets.form_elements.slide
module, 447	module, 455
fobi.contrib.plugins.form_importers.mailchimp_importer_base	fobi.contrib.themes.bootstrap3.widgets.form_elements.slide
module, 448	module, 455
fobi.contrib.plugins.form_importers.mailchimp_importer_forms	fobi.contrib.themes.djangocms_admin_style_theme
module, 448	module, 462
fobi.contrib.plugins.form_importers.mailchimp_importer_forms.apps	fobi.contrib.themes.djangocms_admin_style_theme.apps
module, 449	module, 460
fobi.contrib.plugins.form_importers.mailchimp_importer_views	fobi.contrib.themes.djangocms_admin_style_theme.fobi_theme
module, 450	module, 460
fobi.contrib.themes	fobi.contrib.themes.djangocms_admin_style_theme.widgets
module, 472	module, 460
fobi.contrib.themes.bootstrap3	fobi.contrib.themes.djangocms_admin_style_theme.widgets.f
module, 458	module, 459
fobi.contrib.themes.bootstrap3.apps	fobi.contrib.themes.djangocms_admin_style_theme.widgets.f
module, 456	module, 459
fobi.contrib.themes.bootstrap3.fobi_themes	fobi.contrib.themes.djangocms_admin_style_theme.widgets.f
module, 456	module, 458
fobi.contrib.themes.bootstrap3.widgets	fobi.contrib.themes.djangocms_admin_style_theme.widgets.f
module, 456	module, 459
fobi.contrib.themes.bootstrap3.widgets.form_elements	fobi.contrib.themes.djangocms_admin_style_theme.widgets.f
module, 456	module, 460
fobi.contrib.themes.bootstrap3.widgets.form_elements.contrib.themes.djangocms_admin_style_theme.widgets.f	fobi.contrib.themes.djangocms_admin_style_theme.widgets.f
module, 452	module, 460
fobi.contrib.themes.bootstrap3.widgets.form_elements.contrib.themes.djangocms_admin_style_theme.widgets.f	fobi.contrib.themes.djangocms_admin_style_theme.widgets.f
module, 451	module, 459

<code>fobi.migrations.0002_auto_20150912_1744</code> module, 476	<code>fobi.tests.helpers</code> module, 483
<code>fobi.migrations.0003_auto_20160517_1005</code> module, 476	<code>fobi.tests.test_browser_build_dynamic_forms</code> module, 484
<code>fobi.migrations.0004_auto_20160906_1513</code> module, 476	<code>fobi.tests.test_core</code> module, 485
<code>fobi.migrations.0005_auto_20160908_1457</code> module, 477	<code>fobi.tests.test_drf_integration</code> module, 485
<code>fobi.migrations.0006_auto_20160911_1549</code> module, 477	<code>fobi.tests.test_dynamic_forms</code> module, 486
<code>fobi.migrations.0007_auto_20160926_1652</code> module, 477	<code>fobi.tests.test_feincms_integration</code> module, 486
<code>fobi.migrations.0008_formwizardhandlerentry</code> module, 478	<code>fobi.tests.test_form_importers_mailchimp</code> module, 486
<code>fobi.migrations.0009_formwizardentry_wizard_type</code> module, 478	<code>fobi.tests.test_sortable_dict</code> module, 487
<code>fobi.migrations.0010_formwizardhandler</code> module, 478	<code>fobi.urls</code> module, 487
<code>fobi.migrations.0011_formentry_title</code> module, 478	<code>fobi.urls.edit</code> module, 487
<code>fobi.migrations.0012_auto_20161109_1550</code> module, 479	<code>fobi.urls.view</code> module, 487
<code>fobi.migrations.0013_formwizardentry_show_all_navigation_buttons</code> module, 479	<code>fobi.urls.view</code> module, 487
<code>fobi.migrations.0014_auto_20170516_1413</code> module, 479	<code>fobi.validators</code> module, 554
<code>fobi.models</code> module, 536	<code>fobi.views</code> module, 554
<code>fobi.reusable</code> module, 480	<code>fobi.widgets</code> module, 554
<code>fobi.reusable.markdown_widget</code> module, 480	<code>fobi.wizard</code> module, 493
<code>fobi.reusable.markdown_widget.helpers</code> module, 479	<code>fobi.wizard.views</code> module, 493
<code>fobi.reusable.markdown_widget.widgets</code> module, 480	<code>fobi.wizard.views.dynamic</code> module, 487
<code>fobi.settings</code> module, 550	<code>fobi.wizard.views.views</code> module, 492
<code>fobi.templatetags</code> module, 482	<code>FobiBrowserBuldDynamicFormsTest</code> (class in <i>fobi.tests.test_browser_build_dynamic_forms</i>), 484
<code>fobi.templatetags.fobi_tags</code> module, 480	<code>FobiCoreTest</code> (class in <i>fobi.tests.test_core</i>), 485
<code>fobi.templatetags.future_compat</code> module, 482	<code>FobiDataStructuresTest</code> (class in <i>fobi.tests.test_sortable_dict</i>), 487
<code>fobi.tests</code> module, 487	<code>FobiDjangoRestFrameworkIntegrationTests</code> (class in <i>fobi.tests.test_drf_integration</i>), 485
<code>fobi.tests.base</code> module, 482	<code>FobiDynamicFormsTest</code> (class in <i>fobi.tests.test_dynamic_forms</i>), 486
<code>fobi.tests.constants</code> module, 483	<code>FobiFormEntryViewSet</code> (class in <i>fobi.contrib.apps.drf_integration.views</i>), 322
<code>fobi.tests.core</code> module, 483	<code>FobiFormWidget</code> (class in <i>fobi.contrib.apps.feincms_integration.widgets</i>), 324
<code>fobi.tests.data</code> module, 483	

[FobiFormWidget.Meta](#) (class in [attribute](#)), 355
[fobi.contrib.apps.feincms_integration.widgets](#), 324
[FobiMetaData](#) (class in [form \(fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_element](#)
[attribute\)](#), 355
[fobi.contrib.apps.drf_integration.metadata](#), 321
[form \(fobi.contrib.plugins.form_elements.fields.decimal.base.DecimalInput](#)
[attribute\)](#), 357
[form \(fobi.admin.FormElementEntryInlineAdmin](#) [attribute](#)), 494
[form \(fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_element](#)
[attribute\)](#), 358
[form \(fobi.admin.FormHandlerEntryInlineAdmin](#) [attribute](#)), 496
[form \(fobi.contrib.plugins.form_elements.fields.duration.base.DurationInput](#)
[attribute\)](#), 359
[form \(fobi.admin.FormWizardHandlerEntryInlineAdmin](#)
[attribute\)](#), 497
[form \(fobi.contrib.plugins.form_elements.fields.duration.fobi_form_element](#)
[attribute\)](#), 360
[form \(fobi.base.BasePlugin](#) [attribute](#)), 500
[form \(fobi.contrib.plugins.form_elements.fields.email.base.EmailInputPlug](#)
[attribute\)](#), 362
[form \(fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements.E](#)
[attribute\)](#), 362
[form \(fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin](#)
[attribute\)](#), 364
[form \(fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.File](#)
[attribute\)](#), 365
[form \(fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.File](#)
[attribute\)](#), 365
[form \(fobi.contrib.plugins.form_elements.fields.float.base.FloatInputPlugin](#)
[attribute\)](#), 367
[form \(fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements.Fl](#)
[attribute\)](#), 367
[form \(fobi.contrib.plugins.form_elements.fields.hidden.base.HiddenInputPl](#)
[attribute\)](#), 369
[form \(fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements.](#)
[attribute\)](#), 369
[form \(fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements.In](#)
[attribute\)](#), 371
[form \(fobi.contrib.plugins.form_elements.fields.integer.base.IntegerInputPl](#)
[attribute\)](#), 373
[form \(fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.](#)
[attribute\)](#), 374
[form \(fobi.contrib.plugins.form_elements.fields.ip_address.base.IPAddress](#)
[attribute\)](#), 375
[form \(fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elem](#)
[attribute\)](#), 375
[form \(fobi.contrib.plugins.form_elements.fields.null_boolean.base.NullBoo](#)
[attribute\)](#), 377
[form \(fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_ele](#)
[attribute\)](#), 377
[form \(fobi.contrib.plugins.form_elements.fields.password.fobi_form_elemen](#)
[attribute\)](#), 379
[form \(fobi.contrib.plugins.form_elements.fields.radio.base.RadioInputPlug](#)
[attribute\)](#), 381
[form \(fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements.R](#)
[attribute\)](#), 382
[form \(fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSel](#)
[attribute\)](#), 383
[form \(fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_ele](#)
[attribute\)](#), 383

attribute), 384
 form (fobi.contrib.plugins.form_elements.fields.regex.base.RegexInputPlugin (fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.UR
 attribute), 386
 form (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.RegexInputPlugin (fobi.contrib.plugins.form_elements.security.captcha.base.CaptchaInp
 attribute), 386
 form (fobi.contrib.plugins.form_elements.fields.select.base.SelectInputPlugin (fobi.contrib.plugins.form_elements.security.captcha.fobi_form_eleme
 attribute), 388
 form (fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.SelectInputPlugin (fobi.contrib.plugins.form_elements.security.honeypot.base.Honeypot
 attribute), 389
 form (fobi.contrib.plugins.form_elements.fields.select_model.fobi_form_elements.SelectModelObjectInputPlugin (fobi.contrib.plugins.security.honeypot.fobi_form_eleme
 attribute), 390
 form (fobi.contrib.plugins.form_elements.fields.select_model.fobi_form_elements.SelectModelObjectInputPlugin (fobi.contrib.plugins.base.ReCaptcha
 attribute), 391
 form (fobi.contrib.plugins.form_elements.fields.select_mptt.fobi_form_elements.MPTTModelObjectInputPlugin (fobi.contrib.plugins.security.captcha.fobi_form_eleme
 attribute), 393
 form (fobi.contrib.plugins.form_elements.fields.select_mptt.fobi_form_elements.MPTTModelObjectInputPlugin (fobi.contrib.plugins.base.ReCaptcha
 attribute), 394
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.base.HTTPRepostWiz
 attribute), 396
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.fobi_form_handlers.F
 attribute), 397
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.fobi_form_handlers.F
 attribute), 399
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.fobi_form_handlers.F
 attribute), 400
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.fobi_form_handlers.F
 attribute), 402
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.fobi_form_handlers.F
 attribute), 403
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.fobi_form_handlers.F
 attribute), 404
 form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.MultipleFormHandlers.http_repost.fobi_form_handlers.F
 attribute), 406
 form (fobi.contrib.plugins.form_elements.fields.slider.base.SliderInputPlugin (fobi.contrib.themes.djangocms_admin_style_theme.fobi_theme
 attribute), 408
 form (fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements.SliderInputPlugin (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme
 attribute), 409
 form (fobi.contrib.plugins.form_elements.fields.slug.base.SlugInputPlugin (fobi.contrib.themes.simple.fobi_themes.SimpleTheme
 attribute), 412
 form (fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin (fobi.contrib.plugins.form_handlers.db_store.models
 attribute), 412
 form (fobi.contrib.plugins.form_elements.fields.text.base.TextInputPlugin (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme
 attribute), 413
 form (fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme
 attribute), 414
 form (fobi.contrib.plugins.form_elements.fields.textarea.base.TextareaInputPlugin (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.
 attribute), 415
 form (fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements.TextareaInputPlugin (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme
 attribute), 415
 form (fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme
 attribute), 417
 form (fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin (fobi.contrib.themes.simple.fobi_themes.SimpleTheme
 attribute), 417
 form (fobi.contrib.plugins.form_elements.fields.url.base.URFormPlugin (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.
 attribute), 419

attribute), 461	543
form_edit_ajax (fobi.contrib.themes.simple.fobi_themes.SimpleThemeEntry attribute), 471	545
form_edit_form_entry_option_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457	form_entry (fobi.models.FormWizardFormEntry attribute), 548
form_edit_form_entry_option_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461	form_entry_id (fobi.contrib.apps.feincms_integration.widgets.FobiFormV attribute), 324
form_edit_form_entry_option_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467	form_entry_id (fobi.models.AbstractPluginEntry attribute), 537
form_edit_form_entry_option_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471	form_entry_id (fobi.models.FormFieldsetEntry attribute), 544
form_edit_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461	form_entry_id (fobi.models.FormWizardFormEntry attribute), 548
form_edit_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471	form_entry_submitted_ajax_template (fobi.DjangoCMSAdminStyleTheme attribute), 457
form_element_checkbox_html_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457	form_entry_submitted_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes. attribute), 461
form_element_checkbox_html_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461	form_entry_submitted_ajax_template (fobi.DjangoCMSAdminStyleTheme attribute), 471
form_element_checkbox_html_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467	form_entry_submitted_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
form_element_checkbox_html_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471	form_entry_submitted_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes. attribute), 461
form_element_html_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457	form_entry_submitted_template (fobi.DjangoCMSAdminStyleTheme attribute), 471
form_element_html_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461	form_fieldset_entry (fobi.models.FormElementEntry attribute), 540
form_element_html_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467	form_fieldset_entry_id (fobi.models.FormElementEntry attribute), 540
form_entry (fobi.contrib.apps.feincms_integration.widgets.FobiFormV attribute), 324	form_importer_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
form_entry (fobi.contrib.plugins.form_handlers.db_store.models.SaveFormEntry attribute), 435	form_importers() (in module fobi.context_processors), 518
form_entry (fobi.models.AbstractPluginEntry attribute), 536	form_list (fobi.contrib.plugins.form_importers.mailchimp_importer.views

attribute), 450
 form_list_container_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
 form_list_container_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_list_container_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_list_container_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
 form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_properties_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
 form_properties_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_properties_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_properties_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_properties_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_properties_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_properties_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_properties_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_properties_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_properties_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_radio_element_html_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_radio_element_html_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_sent_get_param (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 324
 form_sent_get_param (fobi.integration.processors.IntegrationProcessor attribute), 472
 form_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
 form_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 attribute), 471
 form_submit_button_text (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 324
 form_template_name (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 324
 form_title (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 324
 form_entry_option_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
 form_view_form_entry_option_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_view_form_entry_option_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_view_form_entry_option_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_view_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_view_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471
 form_wizard_ajax (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
 form_wizard_ajax (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_wizard_entry (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardEntry attribute), 436
 form_wizard_entry (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 461
 form_wizard_entry (fobi.models.FormWizardFormEntry attribute), 548
 form_wizard_entry (fobi.models.FormWizardHandlerEntry attribute), 550
 form_wizard_entry_id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardEntry attribute), 436
 form_wizard_entry_id (fobi.models.AbstractFormWizardPluginEntry attribute), 536
 form_wizard_entry_id (fobi.models.FormWizardFormEntry attribute), 548
 form_wizard_entry_id (fobi.models.FormWizardHandlerEntry attribute), 550
 form_wizard_properties_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 457
 form_wizard_properties_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 467
 form_wizard_properties_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471

(*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 471

form_wizard_snippet_template_name (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 457

form_wizard_snippet_template_name (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 467

form_wizard_snippet_template_name (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 471

form_wizard_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 457

form_wizards_dashboard_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 457

form_wizards_dashboard_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 467

formatted_saved_data() (*fobi.contrib.plugins.form_handlers.db_store.models.AbstractFormEntry* method), 434

FormCallback (class in *fobi.base*), 504

FormCallbackError, 522

FormCallbackRegistry (class in *fobi.base*), 504

FormElement (class in *fobi.models*), 539

FormElement.DoesNotExist, 539

FormElement.MultipleObjectsReturned, 539

FormElementAdmin (class in *fobi.admin*), 493

FormElementEntry (class in *fobi.models*), 540

FormElementEntry.DoesNotExist, 540

FormElementEntry.MultipleObjectsReturned, 540

formelemententry_set (*fobi.models.FormEntry* attribute), 541

formelemententry_set (*fobi.models.FormFieldsetEntry* attribute), 544

FormElementEntryAdmin (class in *fobi.admin*), 494

FormElementEntryAdmin.Meta (class in *fobi.admin*), 494

FormElementEntryForm (class in *fobi.forms*), 526

FormElementEntryForm.Meta (class in *fobi.forms*), 526

FormElementEntryFormSet (in module *fobi.forms*), 526

FormElementEntryInlineAdmin (class in *fobi.admin*), 494

FormElementPlugin (class in *fobi.base*), 505

FormElementPluginDataStorage (class in *fobi.base*), 506

FormElementPluginDoesNotExist, 522

FormElementPluginError, 522

FormElementPluginRegistry (class in *fobi.base*), 506

FormElementPluginWidget (class in *fobi.base*), 506

FormElementPluginWidgetRegistry (class in *fobi.base*), 506

FormEntry (class in *fobi.models*), 541

FormEntry.DoesNotExist, 541

FormEntry.MultipleObjectsReturned, 541

FormEntryAdmin (class in *fobi.admin*), 494

FormEntryAdmin.Meta (class in *fobi.admin*), 494

FormEntryForm (class in *fobi.forms*), 526

FormEntryForm.Meta (class in *fobi.forms*), 526

FormEntrySerializer (class in *fobi.contrib.apps.drf_integration.serializers*), 321

FormEntrySerializer.Meta (class in *fobi.contrib.apps.drf_integration.serializers*), 321

FormFieldPlugin (class in *fobi.base*), 506

FormFieldsetEntry (class in *fobi.models*), 543

FormFieldsetEntry.DoesNotExist, 543

FormFieldsetEntry.MultipleObjectsReturned, 543

formfieldsetentry_set (*fobi.models.FormEntry* attribute), 541

formfieldsetentry_set (*fobi.models.FormFieldsetEntry* attribute), 544

FormFieldsetEntryAdmin (class in *fobi.admin*), 495

FormFieldsetEntryAdmin.Meta (class in *fobi.admin*), 495

FormFieldsetEntryForm (class in *fobi.forms*), 527

FormFieldsetEntryForm.Meta (class in *fobi.forms*), 527

FormHandler (class in *fobi.models*), 544

FormHandler.DoesNotExist, 544

FormHandler.MultipleObjectsReturned, 544

FormHandlerAdmin (class in *fobi.admin*), 495

FormHandlerEntry (class in *fobi.models*), 545

FormHandlerEntry.DoesNotExist, 545

FormHandlerEntry.MultipleObjectsReturned, 545

formhandlerentry_set (*fobi.models.FormEntry* attribute), 541

FormHandlerEntryAdmin (class in *fobi.admin*), 495

FormHandlerEntryAdmin.Meta (class in *fobi.admin*), 495

FormHandlerEntryForm (class in *fobi.forms*), 527

FormHandlerEntryForm.Meta (class in *fobi.forms*), 527

FormHandlerEntryInlineAdmin (class in *fobi.admin*), 496

FormHandlerForm (class in *fobi.forms*), 528

FormHandlerForm.Meta (class in *fobi.forms*), 528

FormHandlerPlugin (class in *fobi.base*), 506

FormHandlerPluginDataStorage (class in *fobi.base*), 507

FormHandlerPluginDoesNotExist, 522

FormHandlerPluginError, 522

FormHandlerPluginRegistry (class in *fobi.base*), 507

FormHandlerPluginWidget (class in *fobi.base*), 508

FormHandlerPluginWidgetRegistry (class in [fobi.models.FormWizardEntry](#) attribute), [fobi.base](#)), 508
 FormImporterPluginRegistry (class in [FormWizardHandlerEntryForm](#) (class in [fobi.forms](#)), [fobi.form_importers](#)), 524
 FormImportersMailchimpTest (class in [FormWizardHandlerEntryForm.Meta](#) (class in [fobi.forms](#)), 529
[fobi.tests.test_form_importers_mailchimp](#)), 486
 FormPluginError, 522
 forms_list_template (class in [FormWizardHandlerEntryForm](#) (class in [fobi.admin](#)), 497
[fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#) (class in [fobi.base](#)), 508
[fobi.contrib.themes.djangoCMS.admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme](#) (class in [fobi.base](#)), 509
[fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#) (class in [fobi.base](#)), 509
[fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#) (class in [fobi.base](#)), 509
[fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#) (class in [fobi.contrib.themes.foundation5.fobi_themes](#)), 466
 FormWizardEntry (class in [fobi.models](#)), 546
 FormWizardEntry.DoesNotExist, 546
 FormWizardEntry.MultipleObjectsReturned, 546
 FormWizardEntryAdmin (class in [fobi.admin](#)), 496
 FormWizardEntryAdmin.Meta (class in [fobi.admin](#)), 496
 FormWizardEntryForm (class in [fobi.forms](#)), 528
 FormWizardEntryForm.Meta (class in [fobi.forms](#)), 528
 FormWizardFormEntry (class in [fobi.models](#)), 548
 FormWizardFormEntry.DoesNotExist, 548
 FormWizardFormEntry.MultipleObjectsReturned, 548
 formwizardformentry_set (class in [fobi.models.FormEntry](#) attribute), 542
 formwizardformentry_set (class in [fobi.models.FormWizardEntry](#) attribute), 546
 FormWizardFormEntryForm (class in [fobi.forms](#)), 528
 FormWizardFormEntryForm.Meta (class in [fobi.forms](#)), 529
 FormWizardFormEntryFormSet (in module [fobi.forms](#)), 529
 FormWizardFormEntryInlineAdmin (class in [fobi.admin](#)), 496
 FormWizardHandler (class in [fobi.models](#)), 549
 FormWizardHandler.DoesNotExist, 549
 FormWizardHandler.MultipleObjectsReturned, 549
 FormWizardHandlerAdmin (class in [fobi.admin](#)), 497
 FormWizardHandlerEntry (class in [fobi.models](#)), 550
 FormWizardHandlerEntry.DoesNotExist, 550
 FormWizardHandlerEntry.MultipleObjectsReturned, 550
 formwizardhandlerentry_set (class in [fobi.models.FormWizardEntry](#) attribute), 546
 generate_ticks() (in module [fobi.contrib.plugins.form_elements.fields.slider.helpers](#)), 411
 get() ([fobi.base.BaseRegistry](#) method), 503
 get() ([fobi.wizard.views.dynamic.DynamicNamedUrlWizardView](#) method), 488
 get() ([fobi.wizard.views.dynamic.DynamicWizardView](#) method), 489
 get() ([fobi.wizard.views.views.CookieWizardView](#) method), 492
 get() ([fobi.wizard.views.views.SessionWizardView](#) method), 492
 get() ([fobi.wizard.views.views.WizardView](#) method), 493
 get_absolute_url() ([fobi.models.FormEntry](#) method), 542
 get_absolute_url() ([fobi.models.FormWizardEntry](#) method), 546
 get_all_cleaned_data() ([fobi.wizard.views.dynamic.DynamicWizardView](#) method), 489
 get_allowed_form_element_plugin_uids() (in module [fobi.utils](#)), 551
 get_allowed_form_handler_plugin_uids() (in module [fobi.utils](#)), 551
 get_allowed_form_wizard_handler_plugin_uids() (in module [fobi.utils](#)), 551
 get_allowed_plugin_uids() (in module [fobi.utils](#)), 551
 get_app_label_and_model_name() (in module [fobi.helpers](#)), 532

method), 282

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.integer.base method), 283

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.integer.base method), 283

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.ip_address method), 284

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.ip_address method), 284

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.decimal(fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean method), 285

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.decimal(fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean method), 285

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.duration(fobi.contrib.apps.drf_integration.form_elements.fields.password method), 286

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.duration(fobi.contrib.apps.drf_integration.form_elements.fields.password method), 287

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.email(fobi.contrib.apps.drf_integration.form_elements.fields.radio.base method), 288

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.email(fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi method), 288

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.file(fobi.contrib.apps.drf_integration.form_elements.fields.range_selector method), 289

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.file(fobi.contrib.apps.drf_integration.form_elements.fields.range_selector method), 289

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.float(fobi.contrib.apps.drf_integration.form_elements.fields.regex.base method), 290

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.float(fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi method), 291

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.hidden(fobi.contrib.apps.drf_integration.form_elements.fields.select.base method), 291

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.hidden(fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi method), 292

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.input(fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple method), 292

get_custom_field_instances()
(fobi.contrib.apps.drf_integration.form_elements.fields.input(fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple method), 293

method), 302

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 303

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 304

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 305

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 305

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 306

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 307

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 308

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 309

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 310

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 310

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 311

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 311

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 312

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 312

get_custom_field_instances() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 313

get_declared_fields() (in module fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, 319

get_field_info() (fobi.contrib.apps.drf_integration.metadata.FobiMetadata, method), 320

get_field_name_to_label_map() (in module fobi.contrib.apps.drf_integration.base), 318

get_fields_from_model_objects() (fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer, method), 321

get_fobi_form_handler_plugin_custom_actions() (in module fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, 480

get_fobi_form_wizard_handler_plugin_custom_actions() (in module fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, 481

get_fobi_plugin() (in module fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, 481

get_form() (fobi.plugin.base.FobiPlugin, method), 500

get_form() (fobi.wizard.views.dynamic.DynamicWizardView, method), 490

get_form_data() (fobi.plugin.base.FobiPlugin, method), 523

get_form_element_entries_for_form_wizard_entry() (fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin.plugin.SelectMultipleInputPlugin, method), 522

get_form_element_plugin_widget() (in module fobi.base), 512

get_form_field_instances() (fobi.base.FormElementPlugin, method), 505

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.base.ContentImagePlugin, method), 327

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 329

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 329

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 331

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 332

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 334

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 335

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 337

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 338

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_image.fobi_fobi_plugin.plugin.ContentImagePlugin, method), 338

(fobi.contrib.plugins.form_elements.content.content_text.base.ContentTextPlugin.form_elements.fields.email.base.EmailInputPlugin), 340

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.fields.email.fobi_form_element_plugin), 341

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_video.base.VideoPlugin.form_elements.fields.file.base.FileInputPlugin), 343

get_form_field_instances() (fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.fields.file.fobi_form_element_plugin), 344

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.boolean.base.BooleanPlugin.form_elements.fields.float.base.FloatInputPlugin), 345

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.fields.float.fobi_form_element_plugin), 346

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.checkbox_select.fobi_form_elements.fields.checkbox_select_multiple_plugin), 347

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.checkbox_select.fobi_form_elements.fields.checkbox_select_multiple_plugin), 348

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin), 350

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.fields.input.fobi_form_element_plugin), 351

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.fields.integer.base.IntegerInputPlugin), 352

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.fields.integer.fobi_form_element_plugin), 353

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.datetime.base.DateTimeInputPlugin), 355

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.fields.ip_address.base.IPAddressPlugin), 355

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.decimal.base.DecimalPlugin), 357

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.fields.null_boolean.base.NullBooleanPlugin), 358

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.duration.base.DurationPlugin), 359

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements.fields.password.base.PasswordInputPlugin), 360

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.email.base.EmailInputPlugin), 362

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.email.fobi_form_element_plugin), 362

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin), 364

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.file.fobi_form_element_plugin), 365

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.float.base.FloatInputPlugin), 367

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.float.fobi_form_element_plugin), 367

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_plugin), 369

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_plugin), 369

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin), 371

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.fields.input.fobi_form_element_plugin), 371

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.fields.integer.fobi_form_element_plugin), 373

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.fields.integer.fobi_form_element_plugin), 374

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.datetime.base.DateTimeInputPlugin), 375

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.fields.ip_address.fobi_form_element_plugin), 375

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.decimal.base.DecimalPlugin), 377

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.fields.null_boolean.fobi_form_element_plugin), 377

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.duration.base.DurationPlugin), 379

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements.fields.password.fobi_form_element_plugin), 379

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.password.base.PasswordInputPlugin), 379

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.password.fobi_form_element_plugin), 379

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.password.fobi_form_element_plugin), 379

(fobi.contrib.plugins.form_elements.security.recaptcha.base.RecaptchaInputPlugin method), 425

get_form_field_instances() (fobi.contrib.plugins.form_elements.security.recaptcha.base.RecaptchaInputPlugin method), 425

get_form_field_instances() (fobi.contrib.plugins.form_elements.test.dummy.base.DummyPlugin method), 427

get_form_field_instances() (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_importers method), 427

get_form_field_type() (in module fobi.templatetags.fobi_tags), 481

get_form_handler_plugin_widget() (in module fobi.base), 512

get_form_hidden_fields_errors() (in module fobi.templatetags.fobi_tags), 481

get_form_importer_plugin_uids() (in module fobi.form_importers), 524

get_form_importer_plugin_urls() (in module fobi.form_importers), 524

get_form_initial() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_form_instance() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_form_kwargs() (fobi.contrib.plugins.form_importers.fobi_form_importers method), 450

get_form_kwargs() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_form_list() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_form_prefix() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_form_step_data() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_form_step_files() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_form_template_choices() (in module fobi.contrib.apps.djangocms_integration.helpers), 272

get_form_template_choices() (in module fobi.contrib.apps.feincms_integration.helpers), 323

get_form_template_choices() (in module fobi.contrib.apps.mezzanine_integration.helpers), 326

get_form_template_name() (fobi.integration.processors.IntegrationProcessor method), 472

get_form_template_name_display() (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method), 472

get_form_wizard_handler_plugin_widget() (in module fobi.base), 512

get_full_name() (fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSelectPlugin method), 531

get_full_path() (fobi.helpers.StrippedRequest method), 531

get_ignorable_form_values() (in module fobi.helpers), 532

get_initial() (fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSelectPlugin method), 384

get_initial() (fobi.contrib.plugins.form_elements.fields.slider.base.SliderPlugin method), 408

get_initial() (fobi.contrib.plugins.form_elements.fields.slider.fobi_form_importers method), 409

get_initial_wizard_data() (fobi.wizard.views.dynamic.DynamicWizardView method), 490

get_initialised_create_form() (fobi.base.BasePlugin method), 500

get_initialised_create_form_or_404() (fobi.base.BasePlugin method), 500

get_initialised_edit_form() (fobi.base.BasePlugin method), 500

get_initialised_edit_form_or_404() (fobi.base.BasePlugin method), 500

get_initialised_edit_form_or_404() (fobi.wizard.views.dynamic.DynamicNamedUrlWizardView class method), 488

get_initialised_edit_form_or_404() (fobi.wizard.views.dynamic.DynamicWizardView class method), 491

get_instance() (fobi.base.BasePlugin method), 500

get_login_required_template_name() (fobi.integration.processors.IntegrationProcessor method), 472

get_model_name_for_object() (in module fobi.helpers), 532

get_next_by_created() (fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedForm method), 434

get_next_by_created() (fobi.contrib.plugins.form_handlers.db_store.models.SavedForm method), 435

get_next_by_created() (fobi.contrib.plugins.form_handlers.db_store.models.SavedForm method), 436

get_next_step() (fobi.wizard.views.dynamic.DynamicWizardView method), 491

get_object() (fobi.contrib.apps.drf_integration.views.FobiFormEntryView method), 322

get_or_create_admin_user() (in module fobi.tests.helpers), 484

get_overridden_form_handlers() (in module fobi.base), 512

fobi.base), 515

`get_registered_plugins()` (*fobi.models.AbstractPluginModel* method), 537

`get_registered_plugins()` (*fobi.models.BaseAbstractPluginEntry* method), 538

`get_registered_plugins()` (*fobi.models.FormElement* method), 539

`get_registered_plugins()` (*fobi.models.FormElementEntry* method), 540

`get_registered_plugins()` (*fobi.models.FormHandler* method), 544

`get_registered_plugins()` (*fobi.models.FormHandlerEntry* method), 546

`get_registered_plugins()` (*fobi.models.FormWizardHandler* method), 549

`get_registered_plugins()` (*fobi.models.FormWizardHandlerEntry* method), 550

`get_registered_plugins()` (in module *fobi.base*), 515

`get_registered_theme_uids()` (in module *fobi.base*), 515

`get_registered_themes()` (in module *fobi.base*), 515

`get_registry()` (*fobi.models.BaseAbstractPluginEntry* method), 538

`get_registry()` (*fobi.models.FormElementEntry* method), 540

`get_registry()` (*fobi.models.FormHandlerEntry* method), 546

`get_registry()` (*fobi.models.FormWizardHandlerEntry* method), 550

`get_rendered_data()` (*fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin* method), 446

`get_rendered_image()` (*fobi.contrib.plugins.form_elements.content.content_image.plugin.ContentImagePlugin* method), 328

`get_rendered_image()` (*fobi.contrib.plugins.form_elements.content.content_image.plugin.ContentImagePlugin* method), 329

`get_rendered_image()` (*fobi.contrib.plugins.form_elements.content.content_image.plugin.ContentImagePlugin* method), 331

`get_rendered_image()` (*fobi.contrib.plugins.form_elements.content.content_image.plugin.ContentImagePlugin* method), 332

`get_rendered_text()` (*fobi.contrib.plugins.form_elements.content.content_markdown.plugin.ContentMarkdownPlugin* method), 334

`get_rendered_text()` (*fobi.contrib.plugins.form_elements.content.content_markdown.plugin.ContentMarkdownPlugin* method), 335

`get_rendered_text()` (*fobi.contrib.plugins.form_elements.content.content_richtext.base.ContentRichtextBase* method), 337

`get_rendered_text()` (*fobi.contrib.plugins.form_elements.content.content_richtext.fobi_form_elements.ContentRichtextFobiFormElement* method), 338

`get_rendered_text()` (*fobi.contrib.plugins.form_elements.content.content_text.base.ContentTextBase* method), 340

`get_rendered_text()` (*fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextFobiFormElement* method), 341

`get_rendered_video()` (*fobi.contrib.plugins.form_elements.content.content_video.base.ContentVideoBase* method), 343

`get_rendered_video()` (*fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.ContentVideoFobiFormElement* method), 344

`get_select_field_choices()` (in module *fobi.helpers*), 533

`get_serializer()` (*fobi.contrib.apps.drf_integration.views.FobiFormEntryView* method), 322

`get_serializer_class()` (*fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet* method), 322

`get_serializer_class()` (in module *fobi.contrib.apps.drf_integration.utils*), 322

`get_setting()` (in module *fobi.conf*), 517

`get_setting()` (in module *fobi.contrib.apps.djangocms_integration.conf*), 271

`get_setting()` (in module *fobi.contrib.apps.feincms_integration.conf*), 323

`get_setting()` (in module *fobi.contrib.apps.mezzanine_integration.conf*), 326

`get_setting()` (in module *fobi.contrib.plugins.form_elements.content.content_image.conf*), 328

`get_setting()` (in module *fobi.contrib.plugins.form_elements.content.content_image_url.conf*), 331

`get_setting()` (in module *fobi.contrib.plugins.form_elements.content.content_text.conf*), 341

`get_setting()` (in module *fobi.contrib.plugins.form_elements.content.content_video.conf*), 343

`get_setting()` (in module *fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf*), 344

348	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.file.conf</i>),	323	get_success_page_template_choices()	(in module <i>fobi.contrib.apps.mezzanine_integration.helpers</i>),
364	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.radio.conf</i>),	326	get_success_page_template_name()	(<i>fobi.integration.processors.IntegrationProcessor</i> method), 473
381	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.range_select.conf</i>),	324	get_success_page_template_name_display()	(<i>fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget</i> method), 324
384	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select.conf</i>),	472	get_template_choices()	(in module <i>fobi.integration.helpers</i>), 472
388	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_model_object_type.conf</i>),	523	get_template_names()	(<i>fobi.form_importers.BaseFormImporter</i> method), 523
391	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_mptt_model_object_type.conf</i>),	515	get_theme()	(in module <i>fobi.base</i>), 515
394	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_mptt_model_object_type.conf</i>),	501	get_updated_plugin_data()	(<i>fobi.base.BasePlugin</i> method), 501
396	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf</i>),	321	get_url()	(<i>fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer</i> method), 321
396	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf</i>),	494	get_urls()	(<i>fobi.admin.FormElementAdmin</i> method), 494
400	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf</i>),	497	get_urls()	(<i>fobi.admin.FormWizardHandlerAdmin</i> method), 497
402	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_multiple_widgets.conf</i>),	551	get_user_form_element_plugin_uids()	(in module <i>fobi.utils</i>), 551
405	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.select_multiple_widgets.conf</i>),	551	get_user_form_element_plugins()	(in module <i>fobi.utils</i>), 551
409	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.fields.slider.conf</i>),	551	get_user_form_element_plugins_grouped()	(in module <i>fobi.utils</i>), 551
423	get_setting()	(in module <i>fobi.contrib.plugins.form_elements.security.honeypot.conf</i>),	551	get_user_form_field_plugin_uids()	(in module <i>fobi.utils</i>), 551
432	get_setting()	(in module <i>fobi.contrib.plugins.form_handlers.db_store.conf</i>),	551	get_user_form_handler_plugin_uids()	(in module <i>fobi.utils</i>), 551
443	get_setting()	(in module <i>fobi.contrib.plugins.form_handlers.mail.conf</i>),	552	get_user_form_handler_plugins()	(in module <i>fobi.utils</i>), 552
531	get_short_name()	(<i>fobi.helpers.StrippedUser</i> method), 531	552	get_user_form_handler_plugins_grouped()	(in module <i>fobi.utils</i>), 552
491	get_step_index()	(<i>fobi.wizard.views.dynamic.DynamicWizardView</i> method), 491	552	get_user_form_wizard_handler_plugin_uids()	(in module <i>fobi.utils</i>), 552
488	get_step_url()	(<i>fobi.wizard.views.dynamic.DynamicNameWizardView</i> method), 488	552	get_user_form_wizard_handler_plugins()	(in module <i>fobi.utils</i>), 552
553	get_success_page_template_choices()	(in module <i>fobi.contrib.apps.djangocms_integration.helpers</i>), 272	552	get_user_form_wizard_handler_plugins_grouped()	(in module <i>fobi.utils</i>), 552
531	get_success_page_template_choices()	(in module <i>fobi.contrib.apps.feincms_integration.helpers</i>), 272	501	get_user_plugin_uids()	(in module <i>fobi.utils</i>), 501
501	get_success_page_template_choices()	(in module <i>fobi.contrib.apps.feincms_integration.helpers</i>), 272	501	get_user_plugins_grouped()	(in module <i>fobi.utils</i>), 501
501	get_success_page_template_choices()	(in module <i>fobi.contrib.apps.feincms_integration.helpers</i>), 272	501	get_username()	(<i>fobi.helpers.StrippedUser</i> method), 531
501	get_success_page_template_choices()	(in module <i>fobi.contrib.apps.feincms_integration.helpers</i>), 272	501	get_widget()	(<i>fobi.base.BasePlugin</i> method), 501
501	get_success_page_template_choices()	(in module <i>fobi.contrib.apps.feincms_integration.helpers</i>), 272	501	get_wizard()	(<i>fobi.form_importers.BaseFormImporter</i> method), 501

615

	attribute), 364	attribute), 394
group (fobi.contrib.plugins.form_elements.fields.file.fobi_form_element_plugin.base.SelectMultiplePlugin)	attribute), 365	attribute), 396
group (fobi.contrib.plugins.form_elements.fields.float.base.FloatInputPlugin)	attribute), 367	attribute), 397
group (fobi.contrib.plugins.form_elements.fields.float.fobi_form_element_plugin.base.SelectMultipleModelObjectPlugin)	attribute), 367	attribute), 399
group (fobi.contrib.plugins.form_elements.fields.hidden.base.HiddenInputPlugin)	attribute), 369	attribute), 400
group (fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_element_plugin.base.SelectMultipleMPTTModelObjectPlugin)	attribute), 369	attribute), 402
group (fobi.contrib.plugins.form_elements.fields.input.base.TextInputPlugin)	attribute), 371	attribute), 403
group (fobi.contrib.plugins.form_elements.fields.input.fobi_form_element_plugin.base.SelectMultipleWithMaxValuePlugin)	attribute), 371	attribute), 404
group (fobi.contrib.plugins.form_elements.fields.integer.base.IntegerInputPlugin)	attribute), 373	attribute), 406
group (fobi.contrib.plugins.form_elements.fields.integer.fobi_form_element_plugin.base.SliderBaseSliderInputPlugin)	attribute), 374	attribute), 408
group (fobi.contrib.plugins.form_elements.fields.ip_address.base.IPAddressInputPlugin)	attribute), 375	attribute), 409
group (fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_element_plugin.base SlugInputPlugin)	attribute), 376	attribute), 412
group (fobi.contrib.plugins.form_elements.fields.null_boolean.base.NullBooleanSelectFieldPlugin)	attribute), 377	attribute), 412
group (fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_element_plugin.base.TextInputPlugin)	attribute), 377	attribute), 413
group (fobi.contrib.plugins.form_elements.fields.password.base.PasswordInputPlugin)	attribute), 379	attribute), 414
group (fobi.contrib.plugins.form_elements.fields.password.fobi_form_element_plugin.base.TextareaPlugin)	attribute), 379	attribute), 415
group (fobi.contrib.plugins.form_elements.fields.radio.base.RadioInputPlugin)	attribute), 381	attribute), 415
group (fobi.contrib.plugins.form_elements.fields.radio.fobi_form_element_plugin.base.TimeInputPlugin)	attribute), 382	attribute), 417
group (fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSelectFormPlugin)	attribute), 384	attribute), 417
group (fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_element_plugin.base URLInputPlugin)	attribute), 384	attribute), 419
group (fobi.contrib.plugins.form_elements.fields.regex.base.RegexFormPlugin)	attribute), 386	attribute), 419
group (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_element_plugin.base.CaptchaIncapchaPlugin)	attribute), 386	attribute), 421
group (fobi.contrib.plugins.form_elements.fields.select.base.SelectFormPlugin)	attribute), 388	attribute), 421
group (fobi.contrib.plugins.form_elements.fields.select.fobi_form_element_plugin.base.HoneypotBaseHoneypotPlugin)	attribute), 389	attribute), 422
group (fobi.contrib.plugins.form_elements.fields.select_model_object_base.SelectModelObjectPlugin)	attribute), 391	attribute), 423
group (fobi.contrib.plugins.form_elements.fields.select_model_object_fobi_form_element_plugin.base.ReCaptchaReCaptchaPlugin)	attribute), 392	attribute), 425
group (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object_base.SelectMPTTModelObjectPlugin)	attribute), 393	attribute), 425
group (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object_fobi_form_element_plugin.base.DummyPlugin)		

- [attribute](#)), 427
- [group](#) (*fobi.contrib.plugins.form_elements.test.dummy.fobi_form_element.DummyPlugin* attribute), 427
- [groups](#) (*fobi.models.AbstractPluginModel* attribute), 537
- [groups](#) (*fobi.models.FormElement* attribute), 539
- [groups](#) (*fobi.models.FormHandler* attribute), 544
- [groups](#) (*fobi.models.FormWizardHandler* attribute), 549
- [groups_list\(\)](#) (*fobi.models.AbstractPluginModel* method), 537
- H**
- [handle\(\)](#) (*fobi.management.commands.fobi_find_broken_entries.Command* method), 473
- [handle\(\)](#) (*fobi.management.commands.fobi_sync_plugins.Command* method), 474
- [handle\(\)](#) (*fobi.management.commands.fobi_update_plugin_data.Command* method), 474
- [handle_uploaded_file\(\)](#) (in module *fobi.helpers*), 534
- [has_add_permission\(\)](#) (*fobi.admin.BasePluginModelAdmin* method), 493
- [has_edit_form_entry_permissions\(\)](#) (in module *fobi.templatetags.fobi_tags*), 481
- [has_value](#) (*fobi.base.FormElementPlugin* attribute), 506
- [has_value](#) (*fobi.base.FormFieldPlugin* attribute), 506
- [has_value](#) (*fobi.base.IntegrationFormElementPlugin* attribute), 510
- [has_value](#) (*fobi.base.IntegrationFormFieldPlugin* attribute), 511
- [has_value](#) (*fobi.base.IntegrationFormHandlerPlugin* attribute), 511
- [has_value\(\)](#) (*fobi.contrib.apps.drf_integration.views.FobiFormEntryView* method), 322
- [help_text](#) (*fobi.base.BaseFormFieldPluginForm* attribute), 498
- [help_text](#) (*fobi.base.BasePlugin* attribute), 501
- [HiddenInputForm](#) (class in *fobi.contrib.plugins.form_elements.fields.hidden.forms*), 370
- [HiddenInputPlugin](#) (class in *fobi.contrib.apps.drf_integration.form_elements.fields.hidden.base*), 291
- [HiddenInputPlugin](#) (class in *fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration_form_elements*), 292
- [HiddenInputPlugin](#) (class in *fobi.contrib.plugins.form_elements.fields.hidden.base*), 369
- [HiddenInputPlugin](#) (class in *fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements*), 369
- [hide_form_title](#) (*fobi.contrib.apps.feincms_integration.widgets.FobiFormTitleWidget* attribute), 337
- [hide_success_page_title](#) (*fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget* attribute), 325
- [HoneypotField](#) (class in *fobi.contrib.plugins.form_elements.security.honeypot.fields*), 423
- [HoneypotInputForm](#) (class in *fobi.contrib.plugins.form_elements.security.honeypot.forms*), 424
- [HoneypotInputPlugin](#) (class in *fobi.contrib.plugins.form_elements.security.honeypot.base*), 422
- [HoneypotInputPlugin](#) (class in *fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements*), 423
- [html_class](#) (*fobi.base.BasePlugin* property), 501
- [html_classes](#) (*fobi.base.BasePlugin* attribute), 501
- [html_classes](#) (*fobi.contrib.plugins.form_elements.content.content_markdown.base* attribute), 334
- [html_classes](#) (*fobi.contrib.plugins.form_elements.content.content_markdown.fobi_integration_form_elements* attribute), 335
- [html_classes](#) (*fobi.contrib.plugins.form_elements.content.content_markdown.fobi_integration_form_elements* attribute), 336
- [html_classes](#) (*fobi.contrib.plugins.form_elements.content.content_richtext.base* attribute), 339
- [html_classes](#) (*fobi.contrib.plugins.form_elements.fields.slider.base.SliderForm* attribute), 408
- [html_classes](#) (*fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements* attribute), 409
- [html_classes](#) (*fobi.contrib.plugins.form_elements.fields.slider.widgets.BaseSliderForm* attribute), 411
- [html_id](#) (*fobi.base.BasePlugin* property), 501
- [HTTPRepostForm](#) (class in *fobi.contrib.plugins.form_handlers.http_repost.forms*), 441
- [HTTPRepostHandlerPlugin](#) (class in *fobi.contrib.apps.drf_integration.form_handlers.http_repost.base*), 315
- [HTTPRepostHandlerPlugin](#) (class in *fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_elements*), 315
- [HTTPRepostHandlerPlugin](#) (class in *fobi.contrib.plugins.form_handlers.http_repost.base*), 438
- [HTTPRepostHandlerPlugin](#) (class in *fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers*), 440
- [HTTPRepostWizardHandlerPlugin](#) (class in *fobi.contrib.plugins.form_handlers.http_repost.base*), 439
- [HTTPRepostWizardHandlerPlugin](#) (class in *fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers*), 439

- 440
- I**
- `id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormDataRow attribute)`, 435
 - `id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry attribute)`, 436
 - `id (fobi.models.FormElement attribute)`, 539
 - `id (fobi.models.FormElementEntry attribute)`, 540
 - `id (fobi.models.FormEntry attribute)`, 542
 - `id (fobi.models.FormFieldsetEntry attribute)`, 544
 - `id (fobi.models.FormHandler attribute)`, 545
 - `id (fobi.models.FormHandlerEntry attribute)`, 546
 - `id (fobi.models.FormWizardEntry attribute)`, 547
 - `id (fobi.models.FormWizardFormEntry attribute)`, 548
 - `id (fobi.models.FormWizardHandler attribute)`, 549
 - `id (fobi.models.FormWizardHandlerEntry attribute)`, 550
 - `import_data() (fobi.form_importers.BaseFormImporter method)`, 523
 - `import_form_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute)`, 461
 - `import_form_entry_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute)`, 467
 - `import_form_entry_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute)`, 471
 - `import_form_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute)`, 461
 - `import_form_entry_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute)`, 467
 - `import_form_entry_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute)`, 471
 - `ImportFormEntryForm (class in fobi.forms)`, 529
 - `ImportFormWizardEntryForm (class in fobi.forms)`, 530
 - `ImproperlyConfigured`, 522
 - `inactive_page_message (fobi.models.FormEntry attribute)`, 542
 - `inactive_page_title (fobi.models.FormEntry attribute)`, 542
 - `initial (fobi.contrib.apps.drf_integration.fields.NoneField attribute)`, 320
 - `initial_dict (fobi.wizard.views.dynamic.DynamicWizardView attribute)`, 491
 - `inlines (fobi.admin.FormEntryAdmin attribute)`, 495
 - `inlines (fobi.admin.FormWizardEntryAdmin attribute)`, 496
 - `input_type (fobi.widgets.NumberInput attribute)`, 554
 - `InputForm (class in fobi.contrib.plugins.form_elements.fields.input.forms)`, 372
 - `InputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.input.base)`, 292
 - `InputPluginDataEntry (class in fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_form_element)`, 293
 - `InputPlugin (class in fobi.contrib.plugins.form_elements.fields.input.base)`, 371
 - `InputPlugin (class in fobi.contrib.plugins.form_elements.fields.input.fobi_form_element)`, 371
 - `insert() (fobi.data_structures.SortableDict method)`, 518
 - `insert_after_key() (fobi.data_structures.SortableDict method)`, 518
 - `insert_before_key() (fobi.data_structures.SortableDict method)`, 518
 - `instance_dict (fobi.wizard.views.dynamic.DynamicWizardView attribute)`, 491
 - `IntegerInputForm (class in fobi.contrib.plugins.form_elements.fields.integer.forms)`, 374
 - `IntegerInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.integer.base)`, 293
 - `IntegerInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_form_element)`, 294
 - `IntegerInputPlugin (class in fobi.contrib.plugins.form_elements.fields.integer.base)`, 373
 - `IntegerInputPlugin (class in fobi.contrib.plugins.form_elements.fields.integer.fobi_form_element)`, 374
 - `integrate_with (fobi.base.IntegrationFormCallback attribute)`, 509
 - `integrate_with (fobi.contrib.apps.drf_integration.form_elements.contenteditable)`, 273
 - `integrate_with (fobi.contrib.apps.drf_integration.form_elements.contenteditable)`, 273
 - `integrate_with (fobi.contrib.apps.drf_integration.form_elements.contenteditable)`, 274
 - `integrate_with (fobi.contrib.apps.drf_integration.form_elements.contenteditable)`, 274
 - `integrate_with (fobi.contrib.apps.drf_integration.form_elements.contenteditable)`, 275
 - `integrate_with (fobi.contrib.apps.drf_integration.form_elements.contenteditable)`, 275
 - `integrate_with (fobi.contrib.apps.drf_integration.form_elements.contenteditable)`, 276

[illegible]

[illegible]

`items()` (*fobi.base.BaseRegistry* method), 504
`items()` (*fobi.data_structures.SortableDict* method), 519
`iterable_to_dict()` (in module *fobi.helpers*), 534

J

`js` (*fobi.reusable.markdown_widget.widgets.MarkdownWidget.Media* attribute), 480
`JSONDataExporter` (class in *fobi.helpers*), 530

K

`keys()` (*fobi.data_structures.SortableDict* method), 519

L

`label` (*fobi.apps.Config* attribute), 498
`label` (*fobi.base.BaseFormFieldPluginForm* attribute), 498
`label` (*fobi.contrib.apps.djangocms_integration.apps.Config* attribute), 271
`label` (*fobi.contrib.apps.drf_integration.apps.Config* attribute), 317
`label` (*fobi.contrib.apps.drf_integration.form_elements.content.content_image.apps.Config* attribute), 272
`label` (*fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.apps.Config* attribute), 273
`label` (*fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.apps.Config* attribute), 275
`label` (*fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.apps.Config* attribute), 276
`label` (*fobi.contrib.apps.drf_integration.form_elements.content.content_text.apps.Config* attribute), 277
`label` (*fobi.contrib.apps.drf_integration.form_elements.content.content_video.apps.Config* attribute), 278
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps.Config* attribute), 279
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.apps.Config* attribute), 280
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.date.apps.Config* attribute), 282
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps.Config* attribute), 283
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps.Config* attribute), 284
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps.Config* attribute), 285
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps.Config* attribute), 286
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.email.apps.Config* attribute), 287
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.file.apps.Config* attribute), 288
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.float.apps.Config* attribute), 290
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps.Config* attribute), 291

`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.input.apps.Config* attribute), 292
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps.Config* attribute), 293
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps.Config* attribute), 294
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps.Config* attribute), 295
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.password.apps.Config* attribute), 296
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps.Config* attribute), 297
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.range_select.apps.Config* attribute), 298
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.regex.apps.Config* attribute), 299
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.select.apps.Config* attribute), 300
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.select_model.apps.Config* attribute), 301
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.apps.Config* attribute), 303
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_image.apps.Config* attribute), 304
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_richtext.apps.Config* attribute), 306
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.slider.apps.Config* attribute), 308
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.slug.apps.Config* attribute), 309
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.text.apps.Config* attribute), 310
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.textarea.apps.Config* attribute), 311
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.time.apps.Config* attribute), 312
`label` (*fobi.contrib.apps.drf_integration.form_elements.fields.url.apps.Config* attribute), 313
`label` (*fobi.contrib.apps.drf_integration.form_handlers.db_store.apps.Config* attribute), 314
`label` (*fobi.contrib.apps.drf_integration.form_handlers.http_repost.apps.Config* attribute), 315
`label` (*fobi.contrib.apps.drf_integration.form_handlers.mail.apps.Config* attribute), 316
`label` (*fobi.contrib.apps.feincms_integration.apps.Config* attribute), 323
`label` (*fobi.contrib.apps.mezzanine_integration.apps.Config* attribute), 325
`label` (*fobi.contrib.plugins.form_elements.content.content_image.apps.Config* attribute), 327
`label` (*fobi.contrib.plugins.form_elements.content.content_image_url.apps.Config* attribute), 330
`label` (*fobi.contrib.plugins.form_elements.content.content_markdown.apps.Config* attribute), 333

label (fobi.contrib.plugins.form_elements.content.content_label(fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model attribute), 337

label (fobi.contrib.plugins.form_elements.content.content_label(fobi.contrib.plugins.form_elements.fields.select_multiple_with_max attribute), 340

label (fobi.contrib.plugins.form_elements.content.content_label(fobi.contrib.plugins.form_elements.fields.slider.apps.Config attribute), 342

label (fobi.contrib.plugins.form_elements.fields.boolean.apps.Config attribute), 345

label (fobi.contrib.plugins.form_elements.fields.checkbox_apps.Config attribute), 347

label (fobi.contrib.plugins.form_elements.fields.date.apps.Config attribute), 350

label (fobi.contrib.plugins.form_elements.fields.date_drop_apps.Config attribute), 352

label (fobi.contrib.plugins.form_elements.fields.datetime.apps.Config attribute), 355

label (fobi.contrib.plugins.form_elements.fields.decimal.apps.Config attribute), 357

label (fobi.contrib.plugins.form_elements.fields.duration.apps.Config attribute), 359

label (fobi.contrib.plugins.form_elements.fields.email.apps.Config attribute), 362

label (fobi.contrib.plugins.form_elements.fields.file.apps.Config attribute), 364

label (fobi.contrib.plugins.form_elements.fields.float.apps.Config attribute), 367

label (fobi.contrib.plugins.form_elements.fields.hidden.apps.Config attribute), 369

label (fobi.contrib.plugins.form_elements.fields.input.apps.Config attribute), 370

label (fobi.contrib.plugins.form_elements.fields.integer.apps.Config attribute), 373

label (fobi.contrib.plugins.form_elements.fields.ip_address_apps.Config attribute), 375

label (fobi.contrib.plugins.form_elements.fields.null_boolean_apps.Config attribute), 377

label (fobi.contrib.plugins.form_elements.fields.password_apps.Config attribute), 379

label (fobi.contrib.plugins.form_elements.fields.radio.apps.Config attribute), 381

label (fobi.contrib.plugins.form_elements.fields.range_select_apps.Config attribute), 383

label (fobi.contrib.plugins.form_elements.fields.regex.apps.Config attribute), 386

label (fobi.contrib.plugins.form_elements.fields.select.apps.Config attribute), 387

label (fobi.contrib.plugins.form_elements.fields.select_model_apps.Config attribute), 390

label (fobi.contrib.plugins.form_elements.fields.select_mptt_apps.Config attribute), 393

label (fobi.contrib.plugins.form_elements.fields.select_multiple_apps.Config attribute), 395

label (fobi.contrib.plugins.form_elements.fields.select_multiple_apps.Config attribute), 399

label (fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model attribute), 401

label (fobi.contrib.plugins.form_elements.fields.select_multiple_with_max attribute), 404

label (fobi.contrib.plugins.form_elements.fields.slider.apps.Config attribute), 408

label (fobi.contrib.plugins.form_elements.fields.slug.apps.Config attribute), 411

label (fobi.contrib.plugins.form_elements.fields.text.apps.Config attribute), 413

label (fobi.contrib.plugins.form_elements.fields.textarea.apps.Config attribute), 415

label (fobi.contrib.plugins.form_elements.fields.time.apps.Config attribute), 417

label (fobi.contrib.plugins.form_elements.fields.url.apps.Config attribute), 418

label (fobi.contrib.plugins.form_elements.security.captcha.apps.Config attribute), 420

label (fobi.contrib.plugins.form_elements.security.honeypot.apps.Config attribute), 422

label (fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config attribute), 425

label (fobi.contrib.plugins.form_elements.test.dummy.apps.Config attribute), 427

label (fobi.contrib.plugins.form_handlers.db_store.apps.Config attribute), 430

label (fobi.contrib.plugins.form_handlers.http_repost.apps.Config attribute), 438

label (fobi.contrib.plugins.form_handlers.mail.apps.Config attribute), 441

label (fobi.contrib.plugins.form_importers.mailchimp_importer.apps.Config attribute), 447

label (fobi.contrib.themes.bootstrap3.apps.Config attribute), 456

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_apps.Config attribute), 451

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_apps.Config attribute), 452

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_apps.Config attribute), 453

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_apps.Config attribute), 453

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_apps.Config attribute), 454

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_apps.Config attribute), 455

label (fobi.contrib.themes.djangocms_admin_style_theme.apps.Config attribute), 460

label (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements_bootstrap3_apps.Config attribute), 458

label (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements_bootstrap3_apps.Config attribute), 459

label (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements_bootstrap3_apps.Config attribute), 466

[map_field_name_to_label\(\)](#) (in module `fobi.contrib.apps.drf_integration.helpers`), 321
[map_field_name_to_label\(\)](#) (in module `fobi.helpers`), 535
[mark_app_setup_as_completed\(\)](#) (in module `fobi.tests.core`), 483
[MarkdownWidget](#) (class in `fobi.reusable.markdown_widget.widgets`), 480
[MarkdownWidget.Media](#) (class in `fobi.reusable.markdown_widget.widgets`), 480
[master_base_template](#) (`fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme` attribute), 457
[master_base_template](#) (`fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme` attribute), 461
[master_base_template](#) (`fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme` attribute), 467
[master_base_template](#) (`fobi.contrib.themes.simple.fobi_themes.SimpleTheme` attribute), 471
[media](#) (`fobi.admin.BasePluginModelAdmin` property), 493
[media](#) (`fobi.admin.FormElementAdmin` property), 494
[media](#) (`fobi.admin.FormElementEntryAdmin` property), 494
[media](#) (`fobi.admin.FormElementEntryInlineAdmin` property), 494
[media](#) (`fobi.admin.FormEntryAdmin` property), 495
[media](#) (`fobi.admin.FormFieldsetEntryAdmin` property), 495
[media](#) (`fobi.admin.FormHandlerAdmin` property), 495
[media](#) (`fobi.admin.FormHandlerEntryAdmin` property), 496
[media](#) (`fobi.admin.FormHandlerEntryInlineAdmin` property), 496
[media](#) (`fobi.admin.FormWizardEntryAdmin` property), 496
[media](#) (`fobi.admin.FormWizardFormEntryInlineAdmin` property), 497
[media](#) (`fobi.admin.FormWizardHandlerAdmin` property), 497
[media](#) (`fobi.admin.FormWizardHandlerEntryInlineAdmin` property), 497
[media](#) (`fobi.contrib.plugins.form_elements.content.content_richtext.forms.ContentRichtextForm` property), 329
[media](#) (`fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm` property), 333
[media](#) (`fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm` property), 336
[media](#) (`fobi.contrib.plugins.form_elements.content.content_richtext.forms.ContentRichtextForm` property), 339
[media](#) (`fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm` property), 342
[media](#) (`fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm` property), 345
[media](#) (`fobi.contrib.plugins.form_elements.fields.boolean.forms.BooleanFieldSelectForm` property), 346
[media](#) (`fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleForm` property), 349
[media](#) (`fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm` property), 352
[media](#) (`fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateDropDownForm` property), 354
[media](#) (`fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeForm` property), 356
[media](#) (`fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm` property), 359
[media](#) (`fobi.contrib.plugins.form_elements.fields.duration.forms.DurationInputForm` property), 361
[media](#) (`fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm` property), 363
[media](#) (`fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm` property), 366
[media](#) (`fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm` property), 368
[media](#) (`fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm` property), 370
[media](#) (`fobi.contrib.plugins.form_elements.fields.input.forms.InputForm` property), 372
[media](#) (`fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm` property), 374
[media](#) (`fobi.contrib.plugins.form_elements.fields.ip_address.forms.IPAddressForm` property), 376
[media](#) (`fobi.contrib.plugins.form_elements.fields.null_boolean.forms.NullBooleanForm` property), 378
[media](#) (`fobi.contrib.plugins.form_elements.fields.password.forms.PasswordForm` property), 380
[media](#) (`fobi.contrib.plugins.form_elements.fields.radio.forms.RadioInputForm` property), 383
[media](#) (`fobi.contrib.plugins.form_elements.fields.range_select.forms.RangeSelectForm` property), 385
[media](#) (`fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm` property), 387
[media](#) (`fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm` property), 390
[media](#) (`fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectForm` property), 392
[media](#) (`fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectForm` property), 395
[media](#) (`fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleForm` property), 398
[media](#) (`fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.forms.SelectMultipleModelObjectForm` property), 401

`multiple_model_objects.forms.SelectMultipleModelFormInputForm`
media (`fobi.forms.ImportFormEntryForm` property), 530
`multiple_model_objects.forms.SelectFormWithMyFormProperty`
property), 530
`media (fobi.fusable.markdown_widget.widgets.MarkdownWidget`
property), 480
`media (fobi.fusable.widgets.BooleanRadioSelect` property), 554
media (`fobi.widgets.NumberInput` property), 554
`media (fobi.widgets.RichSelect` property), 554
media (`fobi.widgets.RichSelectInverseQuotes` property),
`forms.TextareaForm`
media_css (`fobi.base.BasePlugin` attribute), 501
`media_css (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme`
attribute), 457
`media_css (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bo`
attribute), 453
`media_css (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime`
attribute), 454
`media_css (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy`
attribute), 455
`media_css (fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_b`
attribute), 455
`media_css (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes`
attribute), 461
`media_css (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme`
attribute), 468
`media_css (fobi.contrib.themes.foundation5.widgets.form_elements.date_fo`
attribute), 463
`media_css (fobi.contrib.themes.foundation5.widgets.form_elements.datetime`
attribute), 464
`media_css (fobi.contrib.themes.foundation5.widgets.form_elements.dummy`
attribute), 465
`media_css (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_b`
attribute), 465
`media_css (fobi.contrib.themes.foundation5.widgets.form_elements.slider_b`
attribute), 465
`media_css (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes`
attribute), 461
`media_css (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme`
attribute), 468
`media_css (fobi.contrib.themes.foundation5.widgets.form_elements.date_fo`
attribute), 463
`media_css (fobi.contrib.themes.foundation5.widgets.form_elements.datetime`
attribute), 464
`media_js (fobi.contrib.plugins.form_elements.content.content_richtext.wid`
attribute), 339
`media_js (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme`
attribute), 457
`media_js (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bo`
Form attribute), 453
`media_js (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime`
attribute), 454
`media_js (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_b`
attribute), 455
`media_js (fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bo`
attribute), 455
`media_js (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes`
attribute), 461
`media_js (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme`
attribute), 468
`media_js (fobi.contrib.themes.foundation5.widgets.form_elements.date_fo`
attribute), 463
`media_js (fobi.contrib.themes.foundation5.widgets.form_elements.datetime`

media_js (fobi.contrib.themes.foundation5.widgets.form_element attribute), 465	models (fobi.contrib.themes.foundation5.widgets.form_element attribute), 496
media_js (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471	model (fobi.admin.FormWizardFormEntryInlineAdmin attribute), 497
message (fobi.contrib.plugins.form_handlers.mail.fields.MultipleField attribute), 444	model (fobi.admin.FormWizardHandlerEntryInlineAdmin attribute), 497
messages_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 458	model (fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer.Meta attribute), 321
messages_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 462	model (fobi.forms.BulkChangeFormElementPluginsForm.Meta attribute), 524
messages_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 468	model (fobi.forms.BulkChangeFormWizardHandlerPluginsForm.Meta attribute), 525
messages_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471	model (fobi.forms.BulkChangeFormWizardHandlerPluginsForm.Meta attribute), 525
messages_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471	model (fobi.forms.FormElementEntryForm.Meta attribute), 526
messages_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 471	model (fobi.forms.FormEntryForm.Meta attribute), 526
META (fobi.helpers.StrippedRequest property), 530	model (fobi.forms.FormFieldsetEntryForm.Meta attribute), 527
metadata_class (fobi.contrib.apps.drf_integration.views.FormEntryViewsSet attribute), 322	model (fobi.forms.FormFieldsetEntryForm.Meta attribute), 527
Migration (class in fobi.contrib.plugins.form_handlers.db_migration for 0001_initial), 428	model (fobi.forms.FormWizardHandlerForm.Meta attribute), 528
Migration (class in fobi.contrib.plugins.form_handlers.db_migration for 0002_formwizardentry_form), 428	model (fobi.forms.FormWizardFormEntryForm.Meta attribute), 528
Migration (class in fobi.migrations.0001_initial), 474	model (fobi.forms.FormWizardFormEntryForm.Meta attribute), 529
Migration (class in fobi.migrations.0002_auto_20150912_1744), 476	model (fobi.forms.FormWizardHandlerEntryForm.Meta attribute), 529
Migration (class in fobi.migrations.0003_auto_20160517_1005), 476	ModelChoiceField (class in fobi.contrib.apps.drf_integration.fields), 320
Migration (class in fobi.migrations.0004_auto_20160906_1513), 476	MultipleChoiceField (class in fobi.contrib.apps.drf_integration.fields), 320
Migration (class in fobi.migrations.0005_auto_20160908_1452), 477	fobi, 555
Migration (class in fobi.migrations.0006_auto_20160911_1549), 477	fobi.admin, 493
Migration (class in fobi.migrations.0007_auto_20160926_1652), 477	fobi.app, 498
Migration (class in fobi.migrations.0008_formwizardhandlerentry_wizardentry_title), 478	fobi.apps, 498
Migration (class in fobi.migrations.0009_formwizardentry_wizardentry_title), 478	fobi.app, 498
Migration (class in fobi.migrations.0010_formwizardhandler), 478	fobi.conf, 517
Migration (class in fobi.migrations.0011_formentry_title), 478	fobi.constants, 518
Migration (class in fobi.migrations.0012_auto_20161109_1550), 479	fobi.context_processors, 518
Migration (class in fobi.migrations.0013_formwizardentry_show_formwizardentry_title), 479	fobi.contrib, 472
Migration (class in fobi.migrations.0014_auto_20170516_1413), 479	fobi.contrib.apps, 327
model (fobi.admin.FormElementEntryInlineAdmin attribute), 496	fobi.contrib.apps.djangocms_integration, 272
	fobi.contrib.apps.djangocms_integration.apps, 271
	fobi.contrib.apps.djangocms_integration.conf, 271
	fobi.contrib.apps.djangocms_integration.defaults, 271

272	fobi.contrib.apps.drf_integration.form_elements.content
fobi.contrib.apps.djangocms_integration.helpers,	277
272	fobi.contrib.apps.drf_integration.form_elements.content
fobi.contrib.apps.djangocms_integration.settings,	277
272	fobi.contrib.apps.drf_integration.form_elements.content
fobi.contrib.apps.drf_integration,	323
fobi.contrib.apps.drf_integration.apps,	279
317	278
fobi.contrib.apps.drf_integration.base,	fobi.contrib.apps.drf_integration.form_elements.content
317	278
fobi.contrib.apps.drf_integration.dynamic,	fobi.contrib.apps.drf_integration.form_elements.content
319	278
fobi.contrib.apps.drf_integration.fields,	fobi.contrib.apps.drf_integration.form_elements.fields
319	314
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
314	280
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
279	279
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
273	279
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
272	280
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
273	282
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
273	280
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
274	280
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
273	281
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
274	283
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
274	282
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
276	282
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
274	282
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
275	284
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
275	283
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
277	283
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
276	283
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
276	285
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
276	284
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
278	284
fobi.contrib.apps.drf_integration.form_elements,	fobi.contrib.apps.drf_integration.form_elements.fields
277	284

Index 629

Index	631
--------------	------------

fobi.contrib.plugins.form_elements.fields.duration, 359
 fobi.contrib.plugins.form_elements.fields.duration, 359
 fobi.contrib.plugins.form_elements.fields.duration, 360
 fobi.contrib.plugins.form_elements.fields.duration, 361
 fobi.contrib.plugins.form_elements.fields.email, 363
 fobi.contrib.plugins.form_elements.fields.email, 362
 fobi.contrib.plugins.form_elements.fields.email, 362
 fobi.contrib.plugins.form_elements.fields.email, 362
 fobi.contrib.plugins.form_elements.fields.email, 363
 fobi.contrib.plugins.form_elements.fields.file, 367
 fobi.contrib.plugins.form_elements.fields.file, 363
 fobi.contrib.plugins.form_elements.fields.file, 364
 fobi.contrib.plugins.form_elements.fields.file, 364
 fobi.contrib.plugins.form_elements.fields.file, 365
 fobi.contrib.plugins.form_elements.fields.file, 365
 fobi.contrib.plugins.form_elements.fields.file, 366
 fobi.contrib.plugins.form_elements.fields.file, 367
 fobi.contrib.plugins.form_elements.fields.float, 369
 fobi.contrib.plugins.form_elements.fields.float, 367
 fobi.contrib.plugins.form_elements.fields.float, 367
 fobi.contrib.plugins.form_elements.fields.float, 367
 fobi.contrib.plugins.form_elements.fields.float, 367
 fobi.contrib.plugins.form_elements.fields.float, 368
 fobi.contrib.plugins.form_elements.fields.hidden, 370
 fobi.contrib.plugins.form_elements.fields.hidden, 369
 fobi.contrib.plugins.form_elements.fields.hidden, 369
 fobi.contrib.plugins.form_elements.fields.hidden, 369
 fobi.contrib.plugins.form_elements.fields.hidden, 370

fobi.contrib.plugins.form_elements.fields.input, 373
 fobi.contrib.plugins.form_elements.fields.input.apps, 370
 fobi.contrib.plugins.form_elements.fields.input.base, 371
 fobi.contrib.plugins.form_elements.fields.input.constant, 371
 fobi.contrib.plugins.form_elements.fields.input.fobi_f, 371
 fobi.contrib.plugins.form_elements.fields.input.forms, 372
 fobi.contrib.plugins.form_elements.fields.integer, 375
 fobi.contrib.plugins.form_elements.fields.integer.apps, 373
 fobi.contrib.plugins.form_elements.fields.integer.base, 373
 fobi.contrib.plugins.form_elements.fields.integer.fobi, 374
 fobi.contrib.plugins.form_elements.fields.integer.form, 374
 fobi.contrib.plugins.form_elements.fields.ip_address, 377
 fobi.contrib.plugins.form_elements.fields.ip_address.a, 375
 fobi.contrib.plugins.form_elements.fields.ip_address.b, 375
 fobi.contrib.plugins.form_elements.fields.ip_address.f, 375
 fobi.contrib.plugins.form_elements.fields.ip_address.f, 376
 fobi.contrib.plugins.form_elements.fields.null_boolean, 379
 fobi.contrib.plugins.form_elements.fields.null_boolean, 377
 fobi.contrib.plugins.form_elements.fields.null_boolean, 377
 fobi.contrib.plugins.form_elements.fields.null_boolean, 377
 fobi.contrib.plugins.form_elements.fields.null_boolean, 378
 fobi.contrib.plugins.form_elements.fields.password, 381
 fobi.contrib.plugins.form_elements.fields.password.app, 379
 fobi.contrib.plugins.form_elements.fields.password.bas, 379
 fobi.contrib.plugins.form_elements.fields.password.fob, 379
 fobi.contrib.plugins.form_elements.fields.password.for, 380
 fobi.contrib.plugins.form_elements.fields.radio, 383

```

fobi.contrib.plugins.form_elements.fields.radio_fobipson
381                                     390
fobi.contrib.plugins.form_elements.fields.radio_fobbase
381                                     393
fobi.contrib.plugins.form_elements.fields.radio_fobconf
381                                     390
fobi.contrib.plugins.form_elements.fields.radio_fobfilefamily
382                                     390
fobi.contrib.plugins.form_elements.fields.radio_fobfobionform_elements
382                                     391
fobi.contrib.plugins.form_elements.fields.radio_fobforms
382                                     391
fobi.contrib.plugins.form_elements.fields.radio_fobsetting
383                                     391
fobi.contrib.plugins.form_elements.fields.range_fobselect
386                                     392
fobi.contrib.plugins.form_elements.fields.range_fobselectapp
383                                     393
fobi.contrib.plugins.form_elements.fields.range_fobselectbase
383                                     395
fobi.contrib.plugins.form_elements.fields.range_fobselectconf
384                                     393
fobi.contrib.plugins.form_elements.fields.range_fobselectdefault
384                                     393
fobi.contrib.plugins.form_elements.fields.range_fobselectfobipform_elements
384                                     394
fobi.contrib.plugins.form_elements.fields.range_fobselectforms
385                                     394
fobi.contrib.plugins.form_elements.fields.range_fobselectsetting
386                                     394
fobi.contrib.plugins.form_elements.fields.regex_fobi
387                                     395
fobi.contrib.plugins.form_elements.fields.regex_fobipson
386                                     395
fobi.contrib.plugins.form_elements.fields.regex_fobbase
386                                     399
fobi.contrib.plugins.form_elements.fields.regex_fobfobionform_elements
386                                     395
fobi.contrib.plugins.form_elements.fields.regex_fobforms
387                                     396
fobi.contrib.plugins.form_elements.fields.select_fobi
390                                     396
fobi.contrib.plugins.form_elements.fields.select_fobipson
387                                     397
fobi.contrib.plugins.form_elements.fields.select_fobbase
388                                     397
fobi.contrib.plugins.form_elements.fields.select_fobconf
388                                     398
fobi.contrib.plugins.form_elements.fields.select_fobfilefamily
389                                     399
fobi.contrib.plugins.form_elements.fields.select_fobfobionform_elements
389                                     401
fobi.contrib.plugins.form_elements.fields.select_fobforms
389                                     399

```

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slider.constants 399 409

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slider.default 400 409

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slider.fobi_ 400 409

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slider.forms 400 410

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slider.helper 401 411

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slider.settings 401 411

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slider.widget 404 411

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slug, 401 413

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slug.apps, 402 411

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slug.base, 402 412

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slug.fobi_ 403 412

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.slug.fobi_ 403 412

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.text, 403 415

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.text.apps, 404 413

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.text.base, 408 413

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.text.fobi_ 404 414

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.text.forms, 404 414

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.textarea, 405 416

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.textarea.app 405 415

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.textarea.base 405 415

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.textarea.fobi_ 406 415

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.textarea.for 407 416

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.time, 408 418

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.time.apps, 411 416

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.time.base, 408 417

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.time.fobi_ 408 417

fobi.contrib.plugins.form_elements.fields.selector.plugins.form_elements.fields.time.forms, 409 418

[fobi.contrib.plugins.form_elements.fields.url](#), [fobi.contrib.plugins.form_elements.test.dummy.apps](#),
[420](#) [426](#)
[fobi.contrib.plugins.form_elements.fields.url](#), [fobi.contrib.plugins.form_elements.test.dummy.base](#),
[418](#) [427](#)
[fobi.contrib.plugins.form_elements.fields.url](#), [fobi.contrib.plugins.form_elements.test.dummy.fobi_for](#)
[419](#) [427](#)
[fobi.contrib.plugins.form_elements.fields.url](#), [fobi.contrib.plugins.form_elements.test.dummy.widgets](#),
[419](#) [427](#)
[fobi.contrib.plugins.form_elements.fields.url](#), [fobis](#), [fobi.contrib.plugins.form_handlers](#), [447](#)
[419](#) [fobi.contrib.plugins.form_handlers.db_store](#),
[fobi.contrib.plugins.form_elements.security](#), [438](#)
[426](#) [fobi.contrib.plugins.form_handlers.db_store.admin](#),
[fobi.contrib.plugins.form_elements.security.captcha](#), [429](#)
[422](#) [fobi.contrib.plugins.form_handlers.db_store.apps](#),
[fobi.contrib.plugins.form_elements.security.captcha](#), [430](#)
[420](#) [fobi.contrib.plugins.form_handlers.db_store.base](#),
[fobi.contrib.plugins.form_elements.security.captcha](#), [430](#)
[421](#) [fobi.contrib.plugins.form_handlers.db_store.callbacks](#),
[fobi.contrib.plugins.form_elements.security.captcha](#), [432](#) [fobi_form_elements](#),
[421](#) [fobi.contrib.plugins.form_handlers.db_store.conf](#),
[fobi.contrib.plugins.form_elements.security.captcha](#), [432](#) [forms](#),
[421](#) [fobi.contrib.plugins.form_handlers.db_store.defaults](#),
[fobi.contrib.plugins.form_elements.security.honeypot](#), [432](#)
[425](#) [fobi.contrib.plugins.form_handlers.db_store.fobi_form](#)
[fobi.contrib.plugins.form_elements.security.honeypot](#), [432](#) [apps](#),
[422](#) [fobi.contrib.plugins.form_handlers.db_store.helpers](#),
[fobi.contrib.plugins.form_elements.security.honeypot](#), [434](#) [base](#),
[422](#) [fobi.contrib.plugins.form_handlers.db_store.migrations](#)
[fobi.contrib.plugins.form_elements.security.honeypot](#), [439](#) [conf](#),
[423](#) [fobi.contrib.plugins.form_handlers.db_store.migrations](#)
[fobi.contrib.plugins.form_elements.security.honeypot](#), [438](#) [defaults](#),
[423](#) [fobi.contrib.plugins.form_handlers.db_store.migrations](#)
[fobi.contrib.plugins.form_elements.security.honeypot](#), [438](#) [fields](#),
[423](#) [fobi.contrib.plugins.form_handlers.db_store.models](#),
[fobi.contrib.plugins.form_elements.security.honeypot](#), [434](#) [fobi_form_elements](#),
[423](#) [fobi.contrib.plugins.form_handlers.db_store.settings](#),
[fobi.contrib.plugins.form_elements.security.honeypot](#), [437](#) [forms](#),
[424](#) [fobi.contrib.plugins.form_handlers.db_store.urls](#),
[fobi.contrib.plugins.form_elements.security.honeypot](#), [439](#) [settings](#),
[424](#) [fobi.contrib.plugins.form_handlers.db_store.urls.form](#)
[fobi.contrib.plugins.form_elements.security.recaptcha](#), [434](#),
[426](#) [fobi.contrib.plugins.form_handlers.db_store.urls.form](#)
[fobi.contrib.plugins.form_elements.security.recaptcha](#), [434](#) [apps](#),
[425](#) [fobi.contrib.plugins.form_handlers.db_store.views](#),
[fobi.contrib.plugins.form_elements.security.recaptcha](#), [437](#) [base](#),
[425](#) [fobi.contrib.plugins.form_handlers.db_store.widgets](#),
[fobi.contrib.plugins.form_elements.security.recaptcha](#), [434](#) [fobi_form_elements](#),
[425](#) [fobi.contrib.plugins.form_handlers.http_repost](#),
[fobi.contrib.plugins.form_elements.security.recaptcha](#), [434](#) [forms](#),
[426](#) [fobi.contrib.plugins.form_handlers.http_repost.apps](#),
[fobi.contrib.plugins.form_elements.test](#), [438](#)
[428](#) [fobi.contrib.plugins.form_handlers.http_repost.base](#),
[fobi.contrib.plugins.form_elements.test.dummy](#), [438](#)
[428](#) [fobi.contrib.plugins.form_handlers.http_repost.fobi_for](#)

[fobi.migrations.0013_formwizardentry_show_multiplechoicewithmaxfield](#) (class in [fobi.contrib.plugins.form_elements.fields.select_multiple_with_maxfield](#)), 479
[fobi.migrations.0014_auto_20170516_1413](#), 479
[fobi.models](#), 536
[fobi.reusable](#), 480
[fobi.reusable.markdown_widget](#), 480
[fobi.reusable.markdown_widget.helpers](#), 479
[fobi.reusable.markdown_widget.widgets](#), 480
[fobi.settings](#), 550
[fobi.templatetags](#), 482
[fobi.templatetags.fobi_tags](#), 480
[fobi.templatetags.future_compat](#), 482
[fobi.tests](#), 487
[fobi.tests.base](#), 482
[fobi.tests.constants](#), 483
[fobi.tests.core](#), 483
[fobi.tests.data](#), 483
[fobi.tests.helpers](#), 483
[fobi.tests.test_browser_build_dynamic_forms](#), 484
[fobi.tests.test_core](#), 485
[fobi.tests.test_drf_integration](#), 485
[fobi.tests.test_dynamic_forms](#), 486
[fobi.tests.test_feincms_integration](#), 486
[fobi.tests.test_form_importers_mailchimp](#), 486
[fobi.tests.test_sortable_dict](#), 487
[fobi.urls](#), 487
[fobi.urls.edit](#), 487
[fobi.urls.view](#), 487
[fobi.utils](#), 551
[fobi.validators](#), 554
[fobi.views](#), 554
[fobi.widgets](#), 554
[fobi.wizard](#), 493
[fobi.wizard.views](#), 493
[fobi.wizard.views.dynamic](#), 487
[fobi.wizard.views.views](#), 492
[move_after_key\(\)](#) ([fobi.data_structures.SortableDict](#) method), 519
[move_before_key\(\)](#) ([fobi.data_structures.SortableDict](#) method), 519
[MultiEmailField](#) (class in [fobi.contrib.plugins.form_handlers.mail.fields](#)), 444
[MultiEmailWidget](#) (class in [fobi.contrib.plugins.form_handlers.mail.widgets](#)), 447
[MultipleChoiceWithMaxField](#) (class in [fobi.contrib.apps.drf_integration.fields](#)), 320

N

[name](#) ([fobi.apps.Config](#) attribute), 498
[name](#) ([fobi.base.BaseFormFieldPluginForm](#) attribute), 498
[name](#) ([fobi.base.BasePlugin](#) attribute), 501
[name](#) ([fobi.contrib.apps.djangocms_integration.apps.Config](#) attribute), 271
[name](#) ([fobi.contrib.apps.drf_integration.apps.Config](#) attribute), 317
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_image](#) attribute), 272
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_image](#) attribute), 273
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_image](#) attribute), 273
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_image](#) attribute), 273
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_image](#) attribute), 274
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_image](#) attribute), 274
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_manager](#) attribute), 275
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_manager](#) attribute), 275
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_manager](#) attribute), 275
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_rich_text](#) attribute), 276
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_rich_text](#) attribute), 276
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_rich_text](#) attribute), 276
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_text](#) attribute), 277
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_text](#) attribute), 277
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_text](#) attribute), 277
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_video](#) attribute), 278
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_video](#) attribute), 278
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_video](#) attribute), 279
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps.boolean](#) attribute), 279
[name](#) ([fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base.boolean](#) attribute), 279

[illegible]

[illegible]

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 335

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 338

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 337

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 337

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 337

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 338

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 340

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 340

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 341

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 342

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 343

name (fobi.contrib.plugins.form_elements.content.content_max_fields.plugin.fobi_form_element.
attribute), 344

name (fobi.contrib.plugins.form_elements.fields.boolean.apps.Config
attribute), 345

name (fobi.contrib.plugins.form_elements.fields.boolean.base.FloatInputPlugin
attribute), 345

name (fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.
attribute), 346

name (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.plugin.fobi_form_elements.
attribute), 347

name (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.plugin.fobi_form_elements.
attribute), 347

name (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.plugin.fobi_form_elements.
attribute), 348

name (fobi.contrib.plugins.form_elements.fields.date.apps.Config
attribute), 350

name (fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin
attribute), 350

name (fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.
attribute), 351

name (fobi.contrib.plugins.form_elements.fields.date_drop_down.plugin.fobi_form_elements.
attribute), 352

name (fobi.contrib.plugins.form_elements.fields.date_drop_down.plugin.fobi_form_elements.
attribute), 352

name (fobi.contrib.plugins.form_elements.fields.date_drop_down.plugin.fobi_form_elements.
attribute), 353

name (fobi.contrib.plugins.form_elements.fields.datetime.apps.Config
attribute), 355

name (fobi.contrib.plugins.form_elements.fields.datetime.base.IPAddress
attribute), 355

name (fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.
attribute), 356

name (fobi.contrib.plugins.form_elements.fields.decimal.apps.Config
attribute), 357

name (fobi.contrib.plugins.form_elements.fields.decimal.base.NullBoolean
attribute), 357

name (fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config attribute), 378

name (fobi.contrib.plugins.form_elements.fields.null_boolean.select_multiple_model_object attribute), 400

name (fobi.contrib.plugins.form_elements.fields.password.apps.Config attribute), 379

name (fobi.contrib.plugins.form_elements.fields.password.base.PasswordInputPlugin attribute), 401

name (fobi.contrib.plugins.form_elements.fields.password.base.PasswordInputPlugin.select_multiple_model_object attribute), 402

name (fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements attribute), 403

name (fobi.contrib.plugins.form_elements.fields.radio.apps.Config attribute), 381

name (fobi.contrib.plugins.form_elements.fields.radio.base.RadioButton attribute), 404

name (fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements attribute), 404

name (fobi.contrib.plugins.form_elements.fields.range_select.apps.Config attribute), 382

name (fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSlider attribute), 406

name (fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements attribute), 406

name (fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements.Slider attribute), 408

name (fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements.SliderInputPlugin attribute), 408

name (fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements.SliderInputPlugin.select_multiple_model_object attribute), 409

name (fobi.contrib.plugins.form_elements.fields.regex.apps.Config attribute), 386

name (fobi.contrib.plugins.form_elements.fields.regex.base.Regex attribute), 411

name (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements attribute), 412

name (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.RegexInputPlugin attribute), 412

name (fobi.contrib.plugins.form_elements.fields.select.apps.Config attribute), 388

name (fobi.contrib.plugins.form_elements.fields.select.base.Select attribute), 413

name (fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements attribute), 413

name (fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.Select attribute), 414

name (fobi.contrib.plugins.form_elements.fields.select_model_object attribute), 415

name (fobi.contrib.plugins.form_elements.fields.select_model_object.Select attribute), 415

name (fobi.contrib.plugins.form_elements.fields.select_model_object.SelectMultiple attribute), 415

name (fobi.contrib.plugins.form_elements.fields.select_model_object.SelectMultipleModelObject attribute), 415

name (fobi.contrib.plugins.form_elements.fields.select_model_object.SelectMultipleModelObject.fobi_form_elements attribute), 415

name (fobi.contrib.plugins.form_elements.fields.select_mptt.apps.Config attribute), 393

name (fobi.contrib.plugins.form_elements.fields.select_mptt.base.MPTTModelObject attribute), 417

name (fobi.contrib.plugins.form_elements.fields.select_mptt.fobi_form_elements attribute), 417

name (fobi.contrib.plugins.form_elements.fields.select_mptt.fobi_form_elements.MPTTModelObject attribute), 417

name (fobi.contrib.plugins.form_elements.fields.select_multiple.apps.Config attribute), 395

name (fobi.contrib.plugins.form_elements.fields.select_multiple.base.SelectMultiple attribute), 419

name (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements attribute), 419

name (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultiple attribute), 419

name (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultipleModelObject attribute), 419

name (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.URLEncodedForm attribute), 420

name (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.URLEncodedForm.Captcha attribute), 421

name (fobi.contrib.plugins.form_elements.security.captcha.fobi_form_handlers.CaptchaHandlerPlugin.attribute),	421
name (fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_handlers.HoneypotHandlerPlugin.attribute),	422
name (fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_handlers.HoneypotHandlerPlugin.attribute),	422
name (fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_handlers.HoneypotHandlerPlugin.attribute),	423
name (fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_handlers.RecaptchaHandlerPlugin.attribute),	425
name (fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_handlers.RecaptchaHandlerPlugin.attribute),	425
name (fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_handlers.RecaptchaHandlerPlugin.attribute),	425
name (fobi.contrib.plugins.form_elements.test.dummy.apps.Config attribute),	427
name (fobi.contrib.plugins.form_elements.test.dummy.base.DBStoreHandlerPlugin.attribute),	427
name (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_handlers.DBStoreHandlerPlugin.attribute),	427
name (fobi.contrib.plugins.form_handlers.db_store.apps.Config attribute),	430
name (fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin.attribute),	430
name (fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin.attribute),	431
name (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreHandlerPlugin.attribute),	433
name (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreHandlerPlugin.attribute),	433
name (fobi.contrib.plugins.form_handlers.http_repost.apps.Config attribute),	438
name (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandlerPlugin.attribute),	439
name (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandlerPlugin.attribute),	439
name (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers(HTTPRepostHandlerPlugin.attribute),	440
name (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers(HTTPRepostHandlerPlugin.attribute),	440
name (fobi.contrib.plugins.form_handlers.mail.apps.Config attribute),	441
name (fobi.contrib.plugins.form_handlers.mail.base.MailHandlerPlugin.attribute),	442
name (fobi.contrib.plugins.form_handlers.mail.base.MailWidgetPlugin.attribute),	442
name (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers(MailHandlerPlugin.attribute),	444
name (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers(MailWidgetPlugin.attribute),	445
name (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporterFormEntry attribute),	447
name (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.NoDefaultThemeSet attribute),	448

N

- `NoneField` (class in `fobi.contrib.apps.drf_integration.fields.operations`), 477
- `NullBooleanSelectForm` (class in `operations(fobi.migrations.0006_auto_20160911_1549.Migration fobi.contrib.plugins.form_elements.fields.null_boolean.forms)`), 477
- `NullBooleanSelectPlugin` (class in `operations(fobi.migrations.0007_auto_20160926_1652.Migration fobi.contrib.apps.drf_integration.form_elements.fobiforms)`), 477
- `NullBooleanSelectPlugin` (class in `operations(fobi.migrations.0008_formwizardhandlerentry.Migration fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form_elements)`), 478
- `NullBooleanSelectPlugin` (class in `operations(fobi.migrations.0009_formwizardentry_wizard_type.Migration fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form_elements)`), 478
- `NullBooleanSelectPlugin` (class in `operations(fobi.migrations.0010_formwizardhandler.Migration fobi.contrib.plugins.form_elements.fields.null_boolean.fobiforms)`), 478
- `NullBooleanSelectPlugin` (class in `operations(fobi.migrations.0011_formentry_title.Migration fobi.contrib.plugins.form_elements.fields.null_boolean.fobiforms)`), 478
- `NullBooleanSelectPlugin` (class in `operations(fobi.migrations.0012_auto_20161109_1550.Migration fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_integration_form_elements)`), 478
- `NumberInput` (class in `fobi.widgets`), 554

O

- `objects(fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardDataEntry option_template_name attribute)`, 435
- `objects(fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardDataEntry attribute)`, 436
- `objects(fobi.models.FormElement attribute)`, 539
- `objects(fobi.models.FormElementEntry attribute)`, 540
- `objects(fobi.models.FormEntry attribute)`, 542
- `objects(fobi.models.FormFieldsetEntry attribute)`, 544
- `objects(fobi.models.FormHandler attribute)`, 545
- `objects(fobi.models.FormHandlerEntry attribute)`, 546
- `objects(fobi.models.FormWizardEntry attribute)`, 547
- `objects(fobi.models.FormWizardFormEntry attribute)`, 548
- `objects(fobi.models.FormWizardHandler attribute)`, 549
- `objects(fobi.models.FormWizardHandlerEntry attribute)`, 550
- `only_args(fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntryAdmin attribute)`, 429
- `only_args(fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntryAdmin attribute)`, 430
- `operations(fobi.contrib.plugins.form_handlers.db_store.models.PerformsEntryMixin attribute)`, 428
- `operations(fobi.contrib.plugins.form_handlers.db_store.models.PermissionsClass(fobi.contrib.apps.drf_integration.views.FobiFormE attribute)`, 428
- `operations(fobi.migrations.0001_initial.Migration attribute)`, 474
- `operations(fobi.migrations.0002_auto_20150912_1744.Migration attribute)`, 476
- `operations(fobi.migrations.0003_auto_20160517_1005.Migration attribute)`, 476
- `operations(fobi.migrations.0004_auto_20160906_1513.Migration attribute)`, 476

P

- `PasswordInputForm` (class in `fobi.contrib.plugins.form_elements.fields.password.forms`), 380
- `PasswordInputPlugin` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.password.fobiforms)`, 296
- `PasswordInputPlugin` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.password.fobiforms)`, 297
- `PasswordInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.password.base`), 379
- `PasswordInputPluginAdmin` (class in `fobi.contrib.plugins.form_elements.fields.password.fobi_form_element_admin.path(fobi.helpers.StrippedRequest property)`), 531
- `performs_entry_mixin()` (in module `fobi.utils`), 553
- `permissions_class(fobi.contrib.apps.drf_integration.views.FobiFormEntryMixin attribute)`, 322
- `permissions_required()` (in module `fobi.decorators`), 520
- `phantom_js_clean_up()` (in module `fobi.tests.helpers`), 484
- `plugin_data(fobi.models.BaseAbstractPluginEntry attribute)`, 538
- `plugin_data_fields(fobi.base.BaseFormFieldPluginForm attribute)`, 498

plugin_data_fields (fobi.base.BasePluginForm attribute), 503
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 330
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 333
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 336
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 339
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 342
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.slider), 345
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.slug), 346
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.slug), 349
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.textarea), 352
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.time), 354
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 356
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 359
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 361
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 363
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 366
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 368
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 370
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 373
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 375
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 376
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 378
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 380
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 383
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 385
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 387
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 390
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 392
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 395
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 398
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 401
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 403
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.select_multiple), 407
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.slider), 410
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.slug), 413
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.slug), 415
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.textarea), 416
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.time), 418
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 420
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 422
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 424
 plugin_data_fields (fobi.contrib.plugins.form_elements.fields.url), 426
 plugin_data_fields (fobi.contrib.plugins.form_handlers.http_repost), 441
 plugin_data_fields (fobi.contrib.plugins.form_handlers.mail.forms.ManageForm), 446
 plugin_data_fields (fobi.contrib.plugins.form_handlers.mail.forms.ManageForm), 501
 plugin_data_fields (fobi.contrib.plugins.form_handlers.http_repost), 439
 plugin_data_fields (fobi.contrib.plugins.form_handlers.http_repost), 439
 plugin_data_fields (fobi.contrib.plugins.form_handlers.http_repost), 440
 plugin_data_fields (fobi.contrib.plugins.form_handlers.http_repost), 440
 plugin_data_fields (fobi.contrib.plugins.form_handlers.mail.base.MailForm), 442
 plugin_data_fields (fobi.contrib.plugins.form_handlers.mail.base.MailForm), 442
 plugin_data_fields (fobi.contrib.plugins.form_handlers.mail.fobi_form), 444
 plugin_data_fields (fobi.contrib.plugins.form_handlers.mail.fobi_form), 445
 plugin_data_fields (fobi.contrib.plugins.form_handlers.mail.fobi_form), 538

(fobi.contrib.plugins.form_elements.fields.duration.base.DurationInputPlugin method), 359

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements.DurationInputPlugin method), 360

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin method), 364

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.FileInputPlugin method), 365

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_object.base.SelectModelObjectInputPlugin method), 391

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements.SelectModelObjectInputPlugin method), 392

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple.base.SelectMultipleInputPlugin method), 396

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultipleInputPlugin method), 397

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base.SelectMultipleWithMaxInputPlugin method), 404

prepare_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements.SelectMultipleWithMaxInputPlugin method), 406

print_info() (in module fobi.tests.core), 483

process() (fobi.base.BasePlugin method), 502

process() (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method), 325

process_custom_form_field_instance() (fobi.base.IntegrationFormElementPluginProcessor method), 510

process_custom_form_field_instance() (fobi.contrib.apps.drf_integration.base.DRFIntegrationFormElementPluginProcessor method), 317

process_plugin_data() (fobi.base.BasePlugin method), 502

process_step() (fobi.wizard.views.dynamic.DynamicWizardView method), 491

process_step_files() (fobi.wizard.views.dynamic.DynamicWizardView method), 491

Q

quantize() (fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm method), 359

queryset (fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet attribute), 322

Base.DurationInputPlugin

radio_fields (fobi.admin.FormEntryAdmin attribute), 495

radio_fields (fobi.admin.FormWizardEntryAdmin attribute), 496

RadioInputForm (class in fobi.contrib.plugins.form_elements.fields.radio.forms), 382

RadioInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.radio.base), 297

RadioInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_form_elements), 298

RadioInputPlugin (class in fobi.contrib.plugins.form_elements.fields.radio.base), 381

RadioInputPlugin (class in fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements), 382

RangeSelectInputForm (class in fobi.contrib.plugins.form_elements.fields.range_select.forms), 385

RangeSelectInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.range_select), 298

RangeSelectInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.range_select), 299

RangeSelectInputPlugin (class in fobi.contrib.plugins.form_elements.fields.range_select.base), 384

read_only_fields (fobi.contrib.apps.drf_integration.serializers.FormEntryAdminSerializer attribute), 321

readonly_fields (fobi.admin.BasePluginModelAdmin attribute), 493

readonly_fields (fobi.admin.FormElementEntryAdmin attribute), 494

readonly_fields (fobi.admin.FormEntryAdmin attribute), 495

readonly_fields (fobi.admin.FormHandlerEntryAdmin attribute), 496

readonly_fields (fobi.admin.FormWizardEntryAdmin attribute), 496

readonly_fields (fobi.contrib.plugins.form_handlers.db_store.admin.SaveFormHandlerEntryAdmin attribute), 430

ReCaptchaInputForm (class in fobi.contrib.plugins.form_elements.security.recaptcha.forms), 426

ReCaptchaInputPlugin	(class in method), 492	render_revalidation_failure()	(fobi.wizard.views.dynamic.DynamicNamedUrlWizardView method), 488
ReCaptchaInputPlugin	(class in method), 488	render_revalidation_failure()	(fobi.wizard.views.dynamic.DynamicWizardView method), 492
RegexInputForm	(class in method), 492	required	(fobi.base.BaseFormFieldPluginForm attribute), 498
RegexInputPlugin	(class in reset_sequences (fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest method), 482	RichSelect	(class in fobi.widgets), 554
RegexInputPlugin	(class in RichSelectInverseQuotes (class in fobi.widgets), 554	run()	(fobi.base.FormWizardHandlerPlugin method), 507
RegexInputPlugin	(class in 508	run()	(fobi.contrib.apps.drf_integration.form_handlers.db_store.base.DBStoreHandlerPlugin method), 314
RegexInputPlugin	(class in run() (fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration method), 315	run()	(fobi.contrib.apps.drf_integration.form_handlers.http_repost.base.HTTPRepostHandler method), 439
register()	(fobi.base.BaseRegistry method), 504	run()	(fobi.contrib.apps.drf_integration.form_handlers.mail.base.MailHandler method), 316
register()	(fobi.base.FormCallbackRegistry method), 504	run()	(fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_integration method), 316
register()	(fobi.base.IntegrationFormCallbackRegistry method), 510	run()	(fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin method), 430
registry	(fobi.base.BaseRegistry property), 504	run()	(fobi.contrib.plugins.form_handlers.db_store.base.DBStoreWizardHandler method), 431
registry	(fobi.base.IntegrationFormCallbackRegistry property), 510	run()	(fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreHandler method), 433
render()	(fobi.base.BasePlugin method), 502	run()	(fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandler method), 439
render()	(fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method), 325	run()	(fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.FobiFormWidget method), 440
render()	(fobi.contrib.plugins.form_handlers.mail.widgets.MailFormWidget method), 447	run()	(fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailFormWidget method), 445
render()	(fobi.reusable.markdown_widget.widgets.MarkdownWidget method), 480	run_form_handlers()	(in module fobi.base), 516
render()	(fobi.widgets.RichSelect method), 554	run_form_handlers()	(in module fobi.base), 516
render()	(fobi.wizard.views.dynamic.DynamicWizardView method), 491	run_form_handlers()	(in module fobi.base), 516
render_auth_link()	(in module fobi.templatetags.fobi_tags), 481	run_form_handlers()	(in module fobi.base), 516
render_done()	(fobi.wizard.views.dynamic.DynamicNamedUrlWizardView method), 488	run_form_handlers()	(in module fobi.base), 516
render_done()	(fobi.wizard.views.dynamic.DynamicWizardView method), 492	run_form_handlers()	(in module fobi.base), 516
render_fobi_forms_list()	(in module fobi.templatetags.fobi_tags), 481	run_form_handlers()	(in module fobi.base), 516
render_goto_step()	(fobi.wizard.views.dynamic.DynamicNamedUrlWizardView method), 488	run_form_handlers()	(in module fobi.base), 516
render_goto_step()	(fobi.wizard.views.dynamic.DynamicWizardView method), 492	run_form_handlers()	(in module fobi.base), 516
render_next_step()	(fobi.wizard.views.dynamic.DynamicNamedUrlWizardView method), 488	run_form_handlers()	(in module fobi.base), 516
render_next_step()	(fobi.wizard.views.dynamic.DynamicWizardView method), 492	run_form_handlers()	(in module fobi.base), 516

`fobi.contrib.apps.drf_integration.base`), 318

`run_form_wizard_handlers()` (in module `fobi.base`), 516

`run_integration_handler()` (in module `fobi.base`), 507

`run_validation()` (in module `fobi.contrib.apps.drf_integration.fields.NoneField`), 320

S

`safe_text()` (in module `fobi.helpers`), 535

`save_form_data_entry()` (in module `fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin`), 431

`save_form_data_entry()` (in module `fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreHandlerPlugin`), 433

`save_plugin_data()` (in module `fobi.base.BasePluginForm`), 503

`save_plugin_data()` (in module `fobi.contrib.plugins.form_elements.content_image_forms.ContentImageForm`), 330

`saved_data` (in module `fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataRowEntry`), 434

`SavedFormDataRowEntry` (class in `fobi.contrib.plugins.form_handlers.db_store.models`), 435

`SavedFormDataRowEntry.DoesNotExist`, 435

`SavedFormDataRowEntry.MultipleObjectsReturned`, 435

`savedformdataentry_set` (in module `fobi.models.FormEntry` attribute), 542

`SavedFormDataRowEntryAdmin` (class in `fobi.contrib.plugins.form_handlers.db_store.admin`), 429

`SavedFormDataRowEntryAdmin.Meta` (class in `fobi.contrib.plugins.form_handlers.db_store.admin`), 429

`SavedFormWizardDataRowEntry` (class in `fobi.contrib.plugins.form_handlers.db_store.models`), 435

`SavedFormWizardDataRowEntry.DoesNotExist`, 436

`SavedFormWizardDataRowEntry.MultipleObjectsReturned`, 436

`savedformwizarddataentry_set` (in module `fobi.models.FormWizardEntry` attribute), 547

`SavedFormWizardDataRowEntryAdmin` (class in `fobi.contrib.plugins.form_handlers.db_store.admin`), 429

`SavedFormWizardDataRowEntryAdmin.Meta` (class in `fobi.contrib.plugins.form_handlers.db_store.admin`), 430

`SelectInputForm` (class in `fobi.contrib.plugins.form_elements.fields.select.forms`), 389

`SelectInputPlugin` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select.base`), 300

`SelectInputPlugin` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi`), 301

`SelectInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select.base`), 388

`SelectInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select.fobi_form_element`), 389

`SelectModelObjectInputForm` (class in `fobi.contrib.plugins.form_elements.fields.select_model_object.forms`), 392

`SelectModelObjectInputPlugin` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.forms`), 301

`SelectModelObjectInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select_model_object.forms`), 392

`SelectModelObjectInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select_model_object.forms`), 390

`SelectModelObjectInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select_model_object.forms`), 391

`SelectMPTTModelObjectInputForm` (class in `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms`), 395

`SelectMPTTModelObjectInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms`), 393

`SelectMPTTModelObjectInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms`), 394

`SelectMultipleInputForm` (class in `fobi.contrib.plugins.form_elements.fields.select_multiple.forms`), 398

`SelectMultipleInputPlugin` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.forms`), 303

`SelectMultipleInputPlugin` (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.forms`), 304

`SelectMultipleInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select_multiple.base`), 396

`SelectMultipleInputPlugin` (class in `fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form`), 397

`SelectMultipleModelObjectsInputForm` (class in `fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.forms`), 397

401 method), 486

SelectMultipleModelObjectsInputPlugin (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.base`), 486

305 setUp() (fobi.tests.test_sortable_dict.FobiDataStructuresTest method), 487

SelectMultipleModelObjectsInputPlugin (class in `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.fobi_integration_form_elements`), 487

305 setUpClass() (fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest class method), 483

SelectMultipleModelObjectsInputPlugin (class in `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_widgets`), 483

399 setUpTestData() (fobi.tests.test_drf_integration.FobiDjangoRestFramework class method), 485

SelectMultipleModelObjectsInputPlugin (class in `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_widgets`), 485

400 show_all_navigation_buttons (attribute), 547

SelectMultipleMPTTModelObjectsInputForm (class `SimpleTheme` (class in `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_themes`), 470

403 skip() (in module `fobi.tests.core`), 483

SelectMultipleMPTTModelObjectsInputPlugin (class in `fobi.contrib.plugins.form_elements.fields.slider.input_form_model_object`), 483

402 (class in `fobi.contrib.plugins.form_elements.fields.slider.forms`), 410

SelectMultipleMPTTModelObjectsInputPlugin (class in `fobi.contrib.plugins.form_elements.fields.slider.input_plugin_model_object`), 483

403 (class in `fobi.contrib.apps.drf_integration.form_elements.fields.slider.base`), 308

SelectMultipleWithMaxInputForm (class in `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max_input_plugin`), 308

407 (class in `fobi.contrib.apps.drf_integration.form_elements.fields.slider.fobi_widgets`), 308

SelectMultipleWithMaxInputPlugin (class in `fobi.contrib.apps.drf_integration.form_elements.fields.slider.input_plugin_with_max`), 308

306 (class in `fobi.contrib.plugins.form_elements.fields.slider.base`), 408

SelectMultipleWithMaxInputPlugin (class in `fobi.contrib.apps.drf_integration.form_elements.fields.slider.input_plugin_with_max`), 307

307 (class in `fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements`), 409

SelectMultipleWithMaxInputPlugin (class in `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max_input_plugin_widget`), 409

404 (class in `fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3`), 455

SelectMultipleWithMaxInputPlugin (class in `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max_input_plugin`), 543

406 slug (fobi.models.FormWizardEntry attribute), 547

send_email() (fobi.contrib.plugins.form_handlers.mail.callbacks.SlugInputFormMail (class in `fobi.contrib.plugins.form_elements.fields.slug.forms`), 443

send_email() (fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin method), 446

send_mail() (in module `fobi.contrib.apps.drf_integration.form_elements.fields.slug.base`), 309

446 SlugInputPlugin (class in `fobi.contrib.apps.drf_integration.form_elements.fields.slug.fobi_widgets`), 309

SessionWizardView (class in `fobi.wizard.views.views`), 492

setDefault() (fobi.data_structures.SortableDict method), 519

setUp() (fobi.tests.test_core.FobiCoreTest method), 485

setUp() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkTests (class in `fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements`), 412

setUp() (fobi.tests.test_dynamic_forms.FobiDynamicFormsTest method), 486

setUp() (fobi.tests.test_feincms_integration.FeinCMSIntegrationTests (class in `fobi.contrib.plugins.form_handlers.db_store.callbacks.AutoFormDbStore`), 518

SortableDict (class in `fobi.data_structures`), 518

[attribute](#)), 432
[stage](#) ([fobi.contrib.plugins.form_handlers.mail.callbacks.AutoFormMailCallback](#) attribute), 443
[storage](#) ([fobi.base.BasePlugin](#) attribute), 502
[storage](#) ([fobi.base.FormElementPlugin](#) attribute), 506
[storage](#) ([fobi.base.FormElementPluginWidget](#) attribute), 506
[storage](#) ([fobi.base.FormHandlerPlugin](#) attribute), 507
[storage](#) ([fobi.base.FormHandlerPluginWidget](#) attribute), 508
[storage](#) ([fobi.base.FormWizardHandlerPlugin](#) attribute), 509
[storage](#) ([fobi.base.FormWizardHandlerPluginWidget](#) attribute), 509
[storage](#) ([fobi.base.IntegrationFormElementPlugin](#) attribute), 510
[storage](#) ([fobi.base.IntegrationFormHandlerPlugin](#) attribute), 511
[storage_name](#) ([fobi.wizard.views.dynamic.DynamicCookiesWithoutView](#) attribute), 487
[storage_name](#) ([fobi.wizard.views.dynamic.DynamicNamedUrlCookiesWithoutView](#) attribute), 487
[storage_name](#) ([fobi.wizard.views.dynamic.DynamicNamedUrlSessionsWithoutView](#) attribute), 488
[storage_name](#) ([fobi.wizard.views.dynamic.DynamicSessionsWithoutView](#) attribute), 488
[storage_name](#) ([fobi.wizard.views.dynamic.DynamicWizardView](#) attribute), 492
[StrippedRequest](#) (class in [fobi.helpers](#)), 530
[StrippedUser](#) (class in [fobi.helpers](#)), 531
[submit_plugin_form_data\(\)](#)
 ([fobi.base.FormElementPlugin](#) method), 506
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormElementPlugin](#) method), 317
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple_plugin](#) method), 281
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple_plugin](#) method), 281
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base.DateTimeFieldPlugin](#) method), 286
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.duration.base.DurationPlugin](#) method), 287
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_form_element_plugins.form_elements.fields.file.fobi_form_element_plugins](#) method), 289
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_form_element_plugins.form_elements.fields.file.fobi_form_element_plugins](#) method), 289
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 302
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 302
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 303
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 304
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 305
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 305
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 306
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_plugin](#) method), 307
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_plugin](#) method), 347
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_plugin](#) method), 348
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin](#) method), 350
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.form_elements.fields.date.fobi_form_element_plugins.form_elements.fields.date.fobi_form_element_plugins](#) method), 351
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.form_elements.fields.date.fobi_form_element_plugins.form_elements.fields.date.fobi_form_element_plugins](#) method), 353
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.fobi_integration_fields.datetime.base.DateTimeFieldPlugin](#) method), 353
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.fobi_integration_fields.datetime.base.DateTimeFieldPlugin](#) method), 355
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.fobi_integration_fields.datetime.base.DateTimeFieldPlugin](#) method), 356
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.fobi_integration_fields.duration.base.DurationPlugin](#) method), 359
[submit_plugin_form_data\(\)](#)
 ([fobi.contrib.plugins.fobi_integration_fields.file.fobi_form_element_plugins.form_elements.fields.file.fobi_form_element_plugins](#) method), 360

submit_plugin_form_data()	submit_plugin_form_data()
(fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin method), 364	(fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin method), 417
submit_plugin_form_data()	submit_plugin_form_data()
(fobi.contrib.plugins.form_elements.fields.file.fobi_form_element_plugins method), 365	(fobi.contrib.plugins.form_elements.fields.time.fobi_form_element_plugins method), 417
submit_plugin_form_data()	submit_plugin_form_data() (in module fobi.base),
(fobi.contrib.plugins.form_elements.fields.radio.base.RadioInputPlugin method), 381	submit_plugin_form_data() (in module
submit_plugin_form_data()	fobi.contrib.apps.drf_integration.base), 318
(fobi.contrib.plugins.form_elements.fields.radio.fobi_form_element_plugins method), 382	fobi.contrib.apps.drf_integration.views.fobi_form_entry_view class attribute), 543
submit_plugin_form_data()	success_page_message
(fobi.contrib.plugins.form_elements.fields.select.base.SelectMultipleInputPlugin method), 388	(fobi.models.FormWizardEntry attribute), 547
submit_plugin_form_data()	success_page_template_name
(fobi.contrib.plugins.form_elements.fields.select.fobi_form_element_plugins method), 389	(fobi.contrib.apps.drf_integration.widgets.FobiFormWidget attribute), 325
submit_plugin_form_data()	success_page_text (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 325
(fobi.contrib.plugins.form_elements.fields.select_model_object_base.SelectModelObjectInputPlugin method), 391	success_page_title (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 325
submit_plugin_form_data()	(fobi.contrib.plugins.form_elements.fields.select_model_object_base.SelectModelObjectInputPlugin method), 392
submit_plugin_form_data()	success_page_title (fobi.models.FormWizardEntry attribute), 547
(fobi.contrib.plugins.form_elements.fields.select_mptt_model_object_base.SelectMPTTModelObjectInputPlugin method), 393	suffix (fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet attribute), 323
submit_plugin_form_data()	submit_plugin_form_data() (in module fobi.base), 353
(fobi.contrib.plugins.form_elements.fields.select_symptt_model_object_base.SelectSympttModelObjectInputPlugin method), 394	submit_plugin_form_data() (in module fobi.base), 353
submit_plugin_form_data()	T
(fobi.contrib.plugins.form_elements.fields.select_multiple_base.SelectMultipleInputPlugin method), 396	take_screenshot() (fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest method), 483
submit_plugin_form_data()	tearDown() (fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest method), 483
(fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_element_plugins method), 397	tearDown() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 485
submit_plugin_form_data()	tearDownClass() (fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest class method), 483
(fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects_base.SelectMultipleModelObjectsInputPlugin method), 399	template_name (fobi.widgets.NumberInput attribute), 354
submit_plugin_form_data()	template_name (fobi.widgets.RichSelectInverseQuotes attribute), 555
(fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects_base.SelectMultipleMPTTModelObjectsInputPlugin method), 402	template_name (fobi.widgets.dynamic.DynamicWizardView attribute), 492
submit_plugin_form_data()	templates (fobi.contrib.plugins.form_importers.mailchimp_importer.base.MailchimpImporter attribute), 440
(fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects_base.SelectMultipleMPTTModelObjectsInputPlugin method), 403	templates (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers attribute), 449
submit_plugin_form_data()	templates (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers attribute), 524
(fobi.contrib.plugins.form_elements.fields.select_multiple_with_max_base.SelectMultipleWithMaxInputPlugin method), 404	test_01_assemble_form_class_and_render_form()
submit_plugin_form_data()	(fobi.tests.test_form_elements.SelectMultipleWithMaxInputPlugin method), 486

test_01_get_registered_form_element_plugins()	(fobi.tests.test_core.FobiCoreTest method), 485	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 486
test_01_options_action_public_form()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 485	test_1001_open_dashboard()
test_01_sortable_dict_move_before_key()	(fobi.tests.test_sortable_dict.FobiDataStructuresTest method), 487	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_01_test_mailchimp_importer()	(fobi.tests.test_form_importers_mailchimp.FormImportersMailchimpTest method), 486	test_10_fail_get_action_non_public_form()
test_02_get_registered_form_handler_plugins()	(fobi.tests.test_core.FobiCoreTest method), 485	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 486
test_02_put_action_public_form()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 485	test_2001_add_form()
test_02_sortable_dict_move_after_key()	(fobi.tests.test_sortable_dict.FobiDataStructuresTest method), 487	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_03_fail_put_action_public_form()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 485	test_2002_edit_form()
test_03_get_registered_form_callbacks()	(fobi.tests.test_core.FobiCoreTest method), 485	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_04_fail_options_action_non_public_form()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 485	test_2003_delete_form()
test_04_get_registered_themes()	(fobi.tests.test_core.FobiCoreTest method), 485	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_05_action_url()	(fobi.tests.test_core.FobiCoreTest method), 485	test_2004_submit_form()
test_05_fail_put_action()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 485	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_06_form_entry_get_absolute_url()	(fobi.tests.test_core.FobiCoreTest method), 485	test_3001_add_form_elements()
test_06_options_action_non_public_form_auth_user()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 486	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_07_form_wizard_entry_get_absolute_url()	(fobi.tests.test_core.FobiCoreTest method), 485	test_3002_remove_form_elements()
test_07_put_action_non_public_form_auth_user()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 486	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_08_form_entry_is_active()	(fobi.tests.test_core.FobiCoreTest method), 485	test_3003_edit_form_elements()
test_08_get_action_public_form()	(fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 486	(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
test_09_get_action_non_public_form_auth_user()		test_4001_add_form_handlers()
		(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
		test_4002_remove_form_handlers()
		(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
		test_4003_edit_form_handlers()
		(fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 484
		test_fobi_form_widget_public_form()
		(fobi.tests.test_feincms_integration.FeinCMSIntegrationTest method), 486
		TextareaForm (class in fobi.contrib.plugins.form_elements.fields.textarea.forms), 416
		TextareaPlugin (class in fobi.contrib.plugins.drf_integration.form_elements.fields.textarea.base), 311
		TextareaPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.textarea.forms), 311
		TextareaPlugin (class in fobi.contrib.plugins.form_elements.fields.textarea.base), 415
		TextareaPlugin (class in fobi.contrib.plugins.form_elements.fields.textarea.base), 415

[fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements](#)),
[415](#) [TimeInputPlugin](#) (class in [fobi.contrib.apps.drf_integration.form_elements.fields.time.fobi_i](#)
[fobi.contrib.plugins.form_elements.fields.text.forms](#)), [312](#)
[414](#) [TimeInputPlugin](#) (class in [fobi.contrib.plugins.form_elements.fields.time.base](#)),
[fobi.contrib.apps.drf_integration.form_elements.fields.text.base](#)),
[310](#) [TimeInputPlugin](#) (class in [fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements](#)
[fobi.contrib.apps.drf_integration.form_elements.fields.text.fobi_integration_form_elements](#)),
[310](#) [title](#) ([fobi.models.FormEntry](#) attribute), [543](#)
[TextInputPlugin](#) (class in [title](#) ([fobi.models.FormWizardEntry](#) attribute), [547](#)
[fobi.contrib.plugins.form_elements.fields.text.base.to_internal_value\(\)](#)
[413](#) ([fobi.contrib.apps.drf_integration.fields.MultipleChoiceWithMaxl](#)
[TextInputPlugin](#) (class in [method](#)), [320](#)
[fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements](#)), [414](#)
[fobi.contrib.apps.drf_integration.fields.NoneField](#)
[method](#)), [320](#)
[theme\(\)](#) (in module [fobi.context_processors](#)), [518](#)
[theme_uid](#) ([fobi.contrib.themes.bootstrap3.widgets.form_elements.content.fobi.contrib.plugins.form_widgets.fobi_form_fields.MultiContentFi](#)
[attribute](#)), [452](#) [method](#)), [444](#)
[theme_uid](#) ([fobi.contrib.themes.bootstrap3.widgets.form_elements.content.fobi.contrib.plugins.form_widgets.fobi_form_fields.MultiContentFi](#)
[attribute](#)), [452](#) [to_representation\(\)](#) ([fobi.contrib.themes.bootstrap3_widget.fobi_form_elements.ContentRich](#)
[attribute](#)), [452](#) ([fobi.contrib.apps.drf_integration.fields.NoneField](#)
[theme_uid](#) ([fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DatePluginWidget](#)
[attribute](#)), [453](#) [two_dicts_to_string\(\)](#) (in module [fobi.helpers](#)), [535](#)
[theme_uid](#) ([fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DatePluginWidget](#)
[attribute](#)), [454](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute), [504](#)
[theme_uid](#) ([fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DummyPluginWidget](#)
[attribute](#)), [455](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DummyPluginWidget](#)
[attribute](#)), [455](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DummyPluginWidget](#)
[attribute](#)), [455](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.djangoCMS_admin_style_theme.widgets.form_elements.content_richtext_admin_style_widget.fobi_form](#)
[attribute](#)), [459](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.djangoCMS_admin_style_theme.widgets.form_elements.content_richtext_admin_style_widget.fobi_form](#)
[attribute](#)), [459](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget.fobi_form_elements.ContentRichTextPlu](#)
[attribute](#)), [463](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form_elements.DatePluginWidget](#)
[attribute](#)), [463](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form_elements.DatePluginWidget](#)
[attribute](#)), [464](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_elements.DummyPluginWi](#)
[attribute](#)), [465](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.foundation5.widgets.form_handlers.dummy_foundation5_widget.fobi_form_elements.DbStorePluginV](#)
[attribute](#)), [466](#) [type](#) ([fobi.base.BaseFormRegistryWidget](#) attribute),
[theme_uid](#) ([fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi_form_elements.ContentRichTextPlu](#)
[attribute](#)), [469](#) [uid](#) ([fobi.base.BasePlugin](#) attribute), [502](#)
[theme_uid](#) ([fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements.DbStorePluginWidget](#)
[attribute](#)), [469](#) [uid](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_imag](#)
[attribute](#)), [273](#)
[ThemeDoesNotExist](#), [523](#) [uid](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_imag](#)
[attribute](#)), [273](#)
[TimeInputForm](#) (class in [attribute](#)), [273](#)
[fobi.contrib.plugins.form_elements.fields.time.forms](#)), [418](#) [uid](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_imag](#)
[attribute](#)), [274](#)
[TimeInputPlugin](#) (class in [uid](#) ([fobi.contrib.apps.drf_integration.form_elements.content.content_imag](#)
[attribute](#)), [274](#)

[illegible]

[illegible]

uid (*fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandlerPlugin* (class in *fobi.contrib.plugins.form_elements.fields.url.fobi_url_base* attribute), 439)
uid (*fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandlerPlugin* (class in *fobi.contrib.plugins.form_elements.fields.url.base*), attribute), 439
uid (*fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostHandlerPlugin* (class in *URLInputPlugin* attribute), 440)
uid (*fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostHandlerPlugin* (class in *fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements* attribute), 441)
uid (*fobi.contrib.plugins.form_handlers.mail.base.MailHandlerPlugin* (class in *fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormEntry* attribute), 442)
uid (*fobi.contrib.plugins.form_handlers.mail.base.MailWizardHandlerPlugin* (class in *fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardEntry* attribute), 443)
uid (*fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailHandlerPlugin* (class in *fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardEntry* attribute), 445)
uid (*fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailWizardHandlerPlugin* (class in *fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardEntry* attribute), 445)
uid (*fobi.contrib.plugins.form_importers.mailchimp_importers.BaseMailChimpImporter* (class in *fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormEntry* attribute), 448)
uid (*fobi.contrib.plugins.form_importers.mailchimp_importers.BaseMailChimpImporter* (class in *fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormEntry* attribute), 449)
uid (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* (class in *fobi.models.AbstractPluginModel* attribute), 458)
uid (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* (class in *fobi.models.AbstractPluginModel* attribute), 462)
uid (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* (class in *fobi.models.AbstractPluginModel* attribute), 468)
uid (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* (class in *fobi.models.AbstractPluginModel* attribute), 471)
uid (*fobi.form_importers.BaseFormImporter* attribute), 524
uidfy() (*fobi.base.FormCallbackRegistry* method), 505
uidfy() (*fobi.base.IntegrationFormCallbackRegistry* method), 510
uniquify_sequence() (in module *fobi.helpers*), 535
unregister() (*fobi.base.BaseRegistry* method), 504
update() (*fobi.contrib.apps.drf_integration.views.FobiFormEntryView* method), 323
update() (*fobi.data_structures.SortableDict* method), 519
update_plugin_data() (*fobi.base.BasePlugin* method), 502
update_plugin_data() (in module *fobi.helpers*), 535
updated (*fobi.models.FormEntry* attribute), 543
updated (*fobi.models.FormWizardEntry* attribute), 547
url_exists() (in module *fobi.validators*), 554
url_name (*fobi.wizard.views.dynamic.DynamicNamedUrlWizardView* attribute), 488
URLInputForm (class in *fobi.contrib.plugins.form_elements.fields.url.forms*), 419
URLInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.url_base*), 313
URLInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.url.fobi_url_base*), 313
validate() (*fobi.contrib.plugins.form_elements.fields.select_multiple_widget* method), 405
validate() (*fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField* method), 444
validate_form_element_plugin_uid() (in module *fobi.base*), 516
validate_form_handler_plugin_uid() (in module *fobi.base*), 517
validate_form_wizard_handler_plugin_uid() (in module *fobi.base*), 517
validate_initial_for_choices() (in module *fobi.helpers*), 535
validate_initial_for_multiple_choices() (in module *fobi.helpers*), 535
validate_integration_form_element_plugin_uid() (in module *fobi.base*), 517
validate_integration_form_handler_plugin_uid() (in module *fobi.base*), 517
validate_plugin_data() (*fobi.base.BaseFormFieldPluginForm* method), 498
validate_plugin_data() (*fobi.base.BasePluginForm* method), 503
validate_submit_value_as() (in module *fobi.helpers*), 535
validate_theme_uid() (in module *fobi.base*), 517

`value_for_index()` (*fobi.data_structures.SortableDict* method), 519

`values()` (*fobi.data_structures.SortableDict* method), 519

`view_embed_form_entry_ajax_template` (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 458

`view_entries_icon_class` (*fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStoreWidget* attribute), 438

`view_entries_icon_class` (*fobi.contrib.themes.djangocms_admin_style_theme.widgets.DjangoCMSAdminStyleTheme* attribute), 459

`view_entries_icon_class` (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 466

`view_entries_icon_class` (*fobi.contrib.themes.simple.widgets.form_handlers.db_store.widgets.SimpleFormHandlerWidget* attribute), 469

`view_form_entry_ajax_template` (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 458

`view_form_entry_ajax_template` (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme* attribute), 462

`view_form_entry_ajax_template` (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 468

`view_form_entry_ajax_template` (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 471

`view_form_entry_template` (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 458

`view_form_entry_template` (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme* attribute), 462

`view_form_entry_template` (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 468

`view_form_entry_template` (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 471

`view_form_wizard_entry_ajax_template` (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 458

`view_form_wizard_entry_ajax_template` (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 468

`view_form_wizard_entry_ajax_template` (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 471

`view_form_wizard_entry_template` (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 458

`view_form_wizard_entry_template` (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 468

`view_form_wizard_entry_template` (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 471

`widget` (*fobi.base.BasePlugin* attribute), 502

`widget` (*fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneyPotField* attribute), 444

`widget` (*fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField* attribute), 444

`wizard` (*fobi.contrib.plugins.form_importers.mailchimp_importer.base.MailChimpImporter* attribute), 448

`wizard` (*fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter* attribute), 449

`wizard` (*fobi.form_importers.BaseFormImporter* attribute), 524

`WizardView` (class in *fobi.wizard.views.views*), 492