
django-fobi Documentation

Release 0.7.1

Artur Barseghyan <artur.barseghyan@gmail.com>

September 21, 2016

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	21
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	24
9.1.5	Form element plugin final steps	24
10	Creating a new form handler plugin	25
10.1	Define and register the form handler plugin	25
10.1.1	path/to/sample_mail/fobi_form_handlers.py	25
10.1.2	path/to/sample_mail/forms.py	26
10.1.3	Prioritise the execution order	27
10.1.4	Form handler plugin custom actions	27
10.1.5	Form handler plugin final steps	27
11	Creating a new form importer plugin	29
11.1	Define and register the form importer plugin	29
11.1.1	path/to/sample_importer/fobi_form_importers.py	29
11.1.2	path/to/sample_importer/forms.py	30
11.1.3	path/to/sample_importer/views.py	31

11.1.4	Form importer plugin final steps	33
12	Creating a form callback	35
13	Suggestions	37
13.1	Custom action for the form	37
13.2	When you want to customise too many things	37
14	Theming	39
14.1	Create a new theme	39
14.2	Make changes to an existing theme	42
14.2.1	Directory structure	43
14.2.2	fobi_themes.py	43
14.2.3	templates/override_simple_theme/base_view.html	43
14.2.4	templates/override_simple_theme/snippets/form_ajax.html	44
15	Permissions	45
16	Management commands	47
17	Tuning	49
18	Bundled plugins and themes	51
18.1	Bundled form element plugins	51
18.1.1	Fields	51
18.1.2	Content/presentation	52
18.1.3	Security	52
18.1.4	MPTT fields	52
18.1.5	Test	52
18.2	Bundled form handler plugins	52
18.3	Bundled themes	53
19	HTML5 fields	55
20	Loading initial data using GET arguments	57
21	Dynamic initial values	59
22	Submitted form element plugins values	61
23	Rendering forms using third-party libraries	63
23.1	Using <i>django-crispy-forms</i>	63
23.2	Using <i>django-floppyforms</i>	63
24	Import/export forms	65
25	Available translations	67
26	Debugging	69
27	Troubleshooting	71
28	License	73
29	Support	75
30	Author	77

31 Screenshots	79
31.1 Bootstrap3 theme	79
31.1.1 Dashboard	79
31.1.2 Create a form	80
31.1.3 View/edit form	81
Form elements	81
Form handlers	85
31.2 Simple theme	94
31.2.1 View/edit form	94
32 Documentation	99
32.1 fobi package	99
32.1.1 Subpackages	99
fobi.contrib package	99
Subpackages	99
fobi.contrib.apps package	99
Subpackages	99
fobi.contrib.apps.djangocms_integration package	99
Submodules	99
fobi.contrib.apps.djangocms_integration.apps module	99
fobi.contrib.apps.djangocms_integration.cms_plugins module	99
fobi.contrib.apps.djangocms_integration.conf module	100
fobi.contrib.apps.djangocms_integration.defaults module	100
fobi.contrib.apps.djangocms_integration.helpers module	100
fobi.contrib.apps.djangocms_integration.models module	100
fobi.contrib.apps.djangocms_integration.settings module	100
Module contents	100
fobi.contrib.apps.feincms_integration package	100
Submodules	100
fobi.contrib.apps.feincms_integration.apps module	100
fobi.contrib.apps.feincms_integration.conf module	101
fobi.contrib.apps.feincms_integration.defaults module	101
fobi.contrib.apps.feincms_integration.helpers module	101
fobi.contrib.apps.feincms_integration.settings module	101
fobi.contrib.apps.feincms_integration.widgets module	101
Module contents	103
fobi.contrib.apps.mezzanine_integration package	103
Submodules	103
fobi.contrib.apps.mezzanine_integration.admin module	103
fobi.contrib.apps.mezzanine_integration.apps module	103
fobi.contrib.apps.mezzanine_integration.conf module	103
fobi.contrib.apps.mezzanine_integration.defaults module	103
fobi.contrib.apps.mezzanine_integration.helpers module	103
fobi.contrib.apps.mezzanine_integration.models module	103
fobi.contrib.apps.mezzanine_integration.page_processors module	103
fobi.contrib.apps.mezzanine_integration.settings module	104
Module contents	104
fobi.contrib.apps.wagtail_integration package	104
Submodules	104
fobi.contrib.apps.wagtail_integration.apps module	104
fobi.contrib.apps.wagtail_integration.conf module	104
fobi.contrib.apps.wagtail_integration.defaults module	104
fobi.contrib.apps.wagtail_integration.helpers module	104
fobi.contrib.apps.wagtail_integration.models module	104

fobi.contrib.apps.wagtail_integration.settings module	104
Module contents	104
Module contents	104
fobi.contrib.plugins package	104
Subpackages	104
fobi.contrib.plugins.form_elements package	104
Subpackages	104
fobi.contrib.plugins.form_elements.content package	104
Subpackages	104
fobi.contrib.plugins.form_elements.content.content_image package	104
Submodules	104
fobi.contrib.plugins.form_elements.content.content_image.apps module	105
fobi.contrib.plugins.form_elements.content.content_image.conf module	105
fobi.contrib.plugins.form_elements.content.content_image.defaults module	105
fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements module	105
fobi.contrib.plugins.form_elements.content.content_image.forms module	106
fobi.contrib.plugins.form_elements.content.content_image.helpers module	106
fobi.contrib.plugins.form_elements.content.content_image.settings module	107
Module contents	107
fobi.contrib.plugins.form_elements.content.content_text package	107
Submodules	107
fobi.contrib.plugins.form_elements.content.content_text.apps module	107
fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements module	107
fobi.contrib.plugins.form_elements.content.content_text.forms module	108
Module contents	108
fobi.contrib.plugins.form_elements.content.content_video package	108
Submodules	108
fobi.contrib.plugins.form_elements.content.content_video.apps module	108
fobi.contrib.plugins.form_elements.content.content_video.conf module	109
fobi.contrib.plugins.form_elements.content.content_video.defaults module	109
fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements module	109
fobi.contrib.plugins.form_elements.content.content_video.forms module	109
fobi.contrib.plugins.form_elements.content.content_video.settings module	110
Module contents	110
Module contents	110
fobi.contrib.plugins.form_elements.fields package	110
Subpackages	110
fobi.contrib.plugins.form_elements.fields.boolean package	110
Submodules	110
fobi.contrib.plugins.form_elements.fields.boolean.apps module	110
fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements module	110
fobi.contrib.plugins.form_elements.fields.boolean.forms module	111
Module contents	111
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple package	111
Submodules	111
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps module	111
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf module	111
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.defaults module	111
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements module	111

fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms module	112
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.settings module	112
Module contents	112
fobi.contrib.plugins.form_elements.fields.date package	112
Submodules	112
fobi.contrib.plugins.form_elements.fields.date.apps module	113
fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements module	113
fobi.contrib.plugins.form_elements.fields.date.forms module	113
fobi.contrib.plugins.form_elements.fields.date.widgets module	114
Module contents	114
fobi.contrib.plugins.form_elements.fields.date_drop_down package	114
Submodules	114
fobi.contrib.plugins.form_elements.fields.date_drop_down.apps module	114
fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements module	114
fobi.contrib.plugins.form_elements.fields.date_drop_down.forms module	115
Module contents	115
fobi.contrib.plugins.form_elements.fields.datetime package	115
Submodules	115
fobi.contrib.plugins.form_elements.fields.datetime.apps module	115
fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements module	115
fobi.contrib.plugins.form_elements.fields.datetime.forms module	116
fobi.contrib.plugins.form_elements.fields.datetime.widgets module	116
Module contents	116
fobi.contrib.plugins.form_elements.fields.decimal package	117
Submodules	117
fobi.contrib.plugins.form_elements.fields.decimal.apps module	117
fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements module	117
fobi.contrib.plugins.form_elements.fields.decimal.forms module	117
Module contents	118
fobi.contrib.plugins.form_elements.fields.email package	118
Submodules	118
fobi.contrib.plugins.form_elements.fields.email.apps module	118
fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements module	118
fobi.contrib.plugins.form_elements.fields.email.forms module	118
Module contents	119
fobi.contrib.plugins.form_elements.fields.file package	119
Submodules	119
fobi.contrib.plugins.form_elements.fields.file.apps module	119
fobi.contrib.plugins.form_elements.fields.file.conf module	119
fobi.contrib.plugins.form_elements.fields.file.defaults module	119
fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements module	119
fobi.contrib.plugins.form_elements.fields.file.forms module	120
fobi.contrib.plugins.form_elements.fields.file.settings module	120
Module contents	120
fobi.contrib.plugins.form_elements.fields.float package	120
Submodules	120
fobi.contrib.plugins.form_elements.fields.float.apps module	121
fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements module	121
fobi.contrib.plugins.form_elements.fields.float.forms module	121
Module contents	121
fobi.contrib.plugins.form_elements.fields.hidden package	122
Submodules	122

fobi.contrib.plugins.form_elements.fields.hidden.apps module	122
fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements module . .	122
fobi.contrib.plugins.form_elements.fields.hidden.forms module	122
Module contents	123
fobi.contrib.plugins.form_elements.fields.hidden_model_object package	123
Submodules	123
fobi.contrib.plugins.form_elements.fields.hidden_model_object.apps module . .	123
fobi.contrib.plugins.form_elements.fields.hidden_model_object.conf module . .	123
fobi.contrib.plugins.form_elements.fields.hidden_model_object.defaults module .	123
fobi.contrib.plugins.form_elements.fields.hidden_model_object.fobi_form_elements module	123
fobi.contrib.plugins.form_elements.fields.hidden_model_object.forms module . .	123
fobi.contrib.plugins.form_elements.fields.hidden_model_object.settings module .	123
Module contents	123
fobi.contrib.plugins.form_elements.fields.input package	123
Submodules	123
fobi.contrib.plugins.form_elements.fields.input.apps module	123
fobi.contrib.plugins.form_elements.fields.input.constants module	123
fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements module . . .	124
fobi.contrib.plugins.form_elements.fields.input.forms module	124
Module contents	124
fobi.contrib.plugins.form_elements.fields.integer package	124
Submodules	124
fobi.contrib.plugins.form_elements.fields.integer.apps module	124
fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements module . .	125
fobi.contrib.plugins.form_elements.fields.integer.forms module	125
Module contents	125
fobi.contrib.plugins.form_elements.fields.ip_address package	125
Submodules	125
fobi.contrib.plugins.form_elements.fields.ip_address.apps module	126
fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements module	126
fobi.contrib.plugins.form_elements.fields.ip_address.forms module	126
Module contents	126
fobi.contrib.plugins.form_elements.fields.null_boolean package	126
Submodules	126
fobi.contrib.plugins.form_elements.fields.null_boolean.apps module	126
fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements module	126
fobi.contrib.plugins.form_elements.fields.null_boolean.forms module	127
Module contents	127
fobi.contrib.plugins.form_elements.fields.password package	127
Submodules	127
fobi.contrib.plugins.form_elements.fields.password.apps module	127
fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements module .	127
fobi.contrib.plugins.form_elements.fields.password.forms module	127
Module contents	128
fobi.contrib.plugins.form_elements.fields.radio package	128
Submodules	128
fobi.contrib.plugins.form_elements.fields.radio.apps module	128
fobi.contrib.plugins.form_elements.fields.radio.conf module	128
fobi.contrib.plugins.form_elements.fields.radio.defaults module	129
fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements module . . .	129
fobi.contrib.plugins.form_elements.fields.radio.forms module	129
fobi.contrib.plugins.form_elements.fields.radio.settings module	130

Module contents	130
fobi.contrib.plugins.form_elements.fields.regex package	130
Submodules	130
fobi.contrib.plugins.form_elements.fields.regex.apps module	130
fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements module	130
fobi.contrib.plugins.form_elements.fields.regex.forms module	130
Module contents	131
fobi.contrib.plugins.form_elements.fields.select package	131
Submodules	131
fobi.contrib.plugins.form_elements.fields.select.apps module	131
fobi.contrib.plugins.form_elements.fields.select.conf module	131
fobi.contrib.plugins.form_elements.fields.select.defaults module	132
fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements module	132
fobi.contrib.plugins.form_elements.fields.select.forms module	132
fobi.contrib.plugins.form_elements.fields.select.settings module	133
Module contents	133
fobi.contrib.plugins.form_elements.fields.select_model_object package	133
Submodules	133
fobi.contrib.plugins.form_elements.fields.select_model_object.apps module	133
fobi.contrib.plugins.form_elements.fields.select_model_object.conf module	133
fobi.contrib.plugins.form_elements.fields.select_model_object.defaults module	133
fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements module	133
fobi.contrib.plugins.form_elements.fields.select_model_object.forms module	134
fobi.contrib.plugins.form_elements.fields.select_model_object.settings module	134
Module contents	134
fobi.contrib.plugins.form_elements.fields.select_mptt_model_object package	134
Submodules	134
fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps module	134
fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf module	135
fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults module	135
fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.fobi_form_elements module	135
fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms module	135
fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.settings module	135
Module contents	135
fobi.contrib.plugins.form_elements.fields.select_multiple package	135
Submodules	135
fobi.contrib.plugins.form_elements.fields.select_multiple.apps module	135
fobi.contrib.plugins.form_elements.fields.select_multiple.conf module	136
fobi.contrib.plugins.form_elements.fields.select_multiple.defaults module	136
fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements module	136
fobi.contrib.plugins.form_elements.fields.select_multiple.forms module	136
fobi.contrib.plugins.form_elements.fields.select_multiple.settings module	137
Module contents	137
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects package	137
Submodules	137
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps module	137
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf module	137

fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults	
module	138
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements	
module	138
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms	
module	138
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.settings	
module	138
Module contents	139
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects	
package	139
Submodules	139
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps	
module	139
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf	
module	139
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults	
module	139
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_elements	
module	139
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms	
module	139
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.settings	
module	139
Module contents	140
fobi.contrib.plugins.form_elements.fields.slug package	140
Submodules	140
fobi.contrib.plugins.form_elements.fields.slug.apps module	140
fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements module	140
fobi.contrib.plugins.form_elements.fields.slug.forms module	140
Module contents	141
fobi.contrib.plugins.form_elements.fields.text package	141
Submodules	141
fobi.contrib.plugins.form_elements.fields.text.apps module	141
fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements module	141
fobi.contrib.plugins.form_elements.fields.text.forms module	141
Module contents	142
fobi.contrib.plugins.form_elements.fields.textarea package	142
Submodules	142
fobi.contrib.plugins.form_elements.fields.textarea.apps module	142
fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements module	142
fobi.contrib.plugins.form_elements.fields.textarea.forms module	142
Module contents	143
fobi.contrib.plugins.form_elements.fields.time package	143
Submodules	143
fobi.contrib.plugins.form_elements.fields.time.apps module	143
fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements module	143
fobi.contrib.plugins.form_elements.fields.time.forms module	144
Module contents	144
fobi.contrib.plugins.form_elements.fields.url package	144
Submodules	144
fobi.contrib.plugins.form_elements.fields.url.apps module	144
fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements module	145
fobi.contrib.plugins.form_elements.fields.url.forms module	145

Module contents	145
Module contents	145
fobi.contrib.plugins.form_elements.security package	146
Subpackages	146
fobi.contrib.plugins.form_elements.security.captcha package	146
Submodules	146
fobi.contrib.plugins.form_elements.security.captcha.apps module	146
fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements module	146
fobi.contrib.plugins.form_elements.security.captcha.forms module	146
Module contents	147
fobi.contrib.plugins.form_elements.security.honeypot package	147
Submodules	147
fobi.contrib.plugins.form_elements.security.honeypot.apps module	147
fobi.contrib.plugins.form_elements.security.honeypot.conf module	147
fobi.contrib.plugins.form_elements.security.honeypot.defaults module	147
fobi.contrib.plugins.form_elements.security.honeypot.fields module	147
fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements module	148
fobi.contrib.plugins.form_elements.security.honeypot.forms module	148
fobi.contrib.plugins.form_elements.security.honeypot.settings module	148
Module contents	148
fobi.contrib.plugins.form_elements.security.recaptcha package	149
Submodules	149
fobi.contrib.plugins.form_elements.security.recaptcha.apps module	149
fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements module	149
fobi.contrib.plugins.form_elements.security.recaptcha.forms module	149
Module contents	150
Module contents	150
fobi.contrib.plugins.form_elements.test package	150
Subpackages	150
fobi.contrib.plugins.form_elements.test.dummy package	150
Submodules	150
fobi.contrib.plugins.form_elements.test.dummy.apps module	150
fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements module	150
fobi.contrib.plugins.form_elements.test.dummy.widgets module	150
Module contents	150
Module contents	151
Module contents	151
fobi.contrib.plugins.form_handlers package	151
Subpackages	151
fobi.contrib.plugins.form_handlers.db_store package	151
Submodules	151
fobi.contrib.plugins.form_handlers.db_store.admin module	151
fobi.contrib.plugins.form_handlers.db_store.apps module	151
fobi.contrib.plugins.form_handlers.db_store.conf module	152
fobi.contrib.plugins.form_handlers.db_store.defaults module	152
fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers module	152
fobi.contrib.plugins.form_handlers.db_store.helpers module	153
fobi.contrib.plugins.form_handlers.db_store.models module	153
fobi.contrib.plugins.form_handlers.db_store.settings module	154
fobi.contrib.plugins.form_handlers.db_store.urls module	154
fobi.contrib.plugins.form_handlers.db_store.views module	154
fobi.contrib.plugins.form_handlers.db_store.widgets module	154
Module contents	155
fobi.contrib.plugins.form_handlers.http_repost package	155

Submodules	155
fobi.contrib.plugins.form_handlers.http_repost.apps module	155
fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers module	155
fobi.contrib.plugins.form_handlers.http_repost.forms module	155
Module contents	156
fobi.contrib.plugins.form_handlers.mail package	156
Submodules	156
fobi.contrib.plugins.form_handlers.mail.apps module	156
fobi.contrib.plugins.form_handlers.mail.conf module	156
fobi.contrib.plugins.form_handlers.mail.defaults module	157
fobi.contrib.plugins.form_handlers.mail.fields module	157
fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers module	157
fobi.contrib.plugins.form_handlers.mail.forms module	158
fobi.contrib.plugins.form_handlers.mail.helpers module	158
fobi.contrib.plugins.form_handlers.mail.settings module	159
fobi.contrib.plugins.form_handlers.mail.widgets module	159
Module contents	159
Module contents	159
fobi.contrib.plugins.form_importers package	159
Subpackages	159
fobi.contrib.plugins.form_importers.mailchimp_importer package	159
Submodules	159
fobi.contrib.plugins.form_importers.mailchimp_importer.apps module	159
fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers module	159
fobi.contrib.plugins.form_importers.mailchimp_importer.forms module	160
fobi.contrib.plugins.form_importers.mailchimp_importer.views module	161
Module contents	161
Module contents	161
Module contents	161
fobi.contrib.themes package	161
Subpackages	161
fobi.contrib.themes.bootstrap3 package	161
Subpackages	161
fobi.contrib.themes.bootstrap3.widgets package	161
Subpackages	161
fobi.contrib.themes.bootstrap3.widgets.form_elements package	161
Subpackages	161
fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget package	161
Submodules	161
fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps module	161
fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements module	162
Module contents	162
fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget package	162
Submodules	162
fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps module	162
fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_form_elements module	162
Module contents	162

fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget	
package	162
Submodules	162
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps	
module	163
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements	
module	163
Module contents	163
Module contents	163
Module contents	163
Submodules	163
fobi.contrib.themes.bootstrap3.apps module	163
fobi.contrib.themes.bootstrap3.fobi_themes module	163
Module contents	164
fobi.contrib.themes.djangocms_admin_style_theme package	165
Subpackages	165
fobi.contrib.themes.djangocms_admin_style_theme.widgets package	165
Subpackages	165
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers	
package	165
Subpackages	165
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store	
package	165
Submodules	165
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps	
module	165
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements	
module	165
Module contents	165
Module contents	165
Module contents	165
Submodules	165
fobi.contrib.themes.djangocms_admin_style_theme.apps module	166
fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes module	166
Module contents	167
fobi.contrib.themes.foundation5 package	167
Subpackages	167
fobi.contrib.themes.foundation5.widgets package	167
Subpackages	167
fobi.contrib.themes.foundation5.widgets.form_elements package	167
Subpackages	167
fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget	
package	167
Submodules	167
fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps	
module	168
fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form_elements	
module	168
Module contents	168
fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget	
package	168
Submodules	168
fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps	
module	168

fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.fobi_form_element	
module	168
Module contents	168
fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget	
package	168
Submodules	169
fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps	
module	169
fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_element	
module	169
Module contents	169
Module contents	169
fobi.contrib.themes.foundation5.widgets.form_handlers package	169
Subpackages	169
fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget	
package	169
Submodules	169
fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps	
module	169
fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_element	
module	170
Module contents	170
Module contents	170
Module contents	170
Module contents	170
Submodules	170
fobi.contrib.themes.foundation5.apps module	170
fobi.contrib.themes.foundation5.fobi_themes module	170
Module contents	171
fobi.contrib.themes.simple package	171
Subpackages	171
fobi.contrib.themes.simple.widgets package	171
Subpackages	171
fobi.contrib.themes.simple.widgets.form_handlers package	171
Subpackages	172
fobi.contrib.themes.simple.widgets.form_handlers.db_store package	172
Submodules	172
fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps module	172
fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements	
module	172
Module contents	172
Module contents	172
Module contents	172
Module contents	172
Submodules	172
fobi.contrib.themes.simple.apps module	172
fobi.contrib.themes.simple.fobi_themes module	173
Module contents	174
Module contents	174
Module contents	174
fobi.integration package	174
Submodules	174
fobi.integration.helpers module	174
fobi.integration.processors module	174
Module contents	175
fobi.management package	175

Subpackages	175
fobi.management.commands package	175
Submodules	175
fobi.management.commands.fobi_find_broken_entries module	175
fobi.management.commands.fobi_migrate_03_to_04 module	175
fobi.management.commands.fobi_sync_plugins module	176
fobi.management.commands.fobi_update_plugin_data module	176
Module contents	176
Module contents	176
fobi.templatetags package	176
Submodules	176
fobi.templatetags.fobi_tags module	176
fobi.templatetags.future_compat module	177
Module contents	178
fobi.tests package	178
Submodules	178
fobi.tests.base module	178
fobi.tests.constants module	178
fobi.tests.data module	178
fobi.tests.helpers module	178
fobi.tests.test_browser_build_dynamic_forms module	179
fobi.tests.test_core module	180
fobi.tests.test_dynamic_forms module	180
fobi.tests.test_form_importers_mailchimp module	180
fobi.tests.test_sortable_dict module	180
Module contents	181
fobi.urls package	181
Submodules	181
fobi.urls.edit module	181
fobi.urls.view module	181
Module contents	181
32.1.2 Submodules	181
32.1.3 fobi.admin module	181
32.1.4 fobi.app module	184
32.1.5 fobi.apps module	184
32.1.6 fobi.base module	184
32.1.7 fobi.compat module	197
32.1.8 fobi.conf module	199
32.1.9 fobi.constants module	199
32.1.10 fobi.context_processors module	199
32.1.11 fobi.data_structures module	200
32.1.12 fobi.decorators module	201
32.1.13 fobi.defaults module	202
32.1.14 fobi.discover module	202
32.1.15 fobi.dynamic module	203
32.1.16 fobi.exceptions module	203
32.1.17 fobi.form_importers module	204
32.1.18 fobi.form_utils module	205
32.1.19 fobi.forms module	205
32.1.20 fobi.helpers module	207
32.1.21 fobi.models module	211
32.1.22 fobi.settings module	221
32.1.23 fobi.test module	222
32.1.24 fobi.utils module	222

32.1.25	fobi.validators module	223
32.1.26	fobi.views module	224
32.1.27	fobi.widgets module	227
32.1.28	Module contents	227
32.2	Quick start	227
32.2.1	Part 1: standard Django installation	227
	Installation and configuration	227
	Install the package in your environment.	227
	INSTALLED_APPS	227
	TEMPLATE_CONTEXT_PROCESSORS	229
	urlpatterns	230
	Update the database	230
	Specify the active theme	230
	Permissions	230
32.2.2	Part 2: Integration with DjangoCMS	231
33	Indices and tables	233
	Bibliography	235
	Python Module Index	237

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 1.5, 1.6, 1.7, 1.8, 1.9
- Python $\geq 2.6.8$, ≥ 2.7 , ≥ 3.3

Note, that Django 1.10 is not yet proclaimed to be flawlessly supported, however it's in progress. The latest core and contrib packages (from master branch, with no additional dependencies) have been tested against the latest stable Django 1.10 release. All tests have successfully passed, although it's yet too early to claim that Django 1.10 is fully supported.

Key concepts

- Each form consists of elements. Form elements are divided into two groups:
 1. form fields (input field, textarea, hidden field, file field, etc.).
 2. content (presentational) elements (text, image, embed video, 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.

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.
- Anti-spam solutions like *CAPTCHA*, *ReCAPTCHA* or *Honeypot* come out of the box (*CAPTCHA* and *ReCAPTCHA* do require additional third-party apps to be installed).
- 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 repost 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.
- *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 *djangoCMS-admin-style*).
- 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:

- Integration with *django-rest-framework* (in version 0.8).
- Fieldsets (in version 0.9).

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:

- [PythonHosted](#)
- [ReadTheDocs](#)

6.1 Live demo

See the [live demo app](#) on Heroku.

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 -e git+https://github.com/barseghyanartur/django-fobi@stable#egg=django-fobi
```

Or latest stable version from BitBucket:

```
pip install -e hg+https://bitbucket.org/barseghyanartur/django-fobi@stable#egg=django-fobi
```

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',
'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_text',
'fobi.contrib.plugins.form_elements.content.content_video',

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

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

3. Make appropriate changes to the `TEMPLATE_CONTEXT_PROCESSORS` of the your projects' Django settings.

And the following to the context processors.

```
TEMPLATE_CONTEXT_PROCESSORS = (
    # ...
    "fobi.context_processors.theme",
    # ...
)
```

Make sure that `django.core.context_processors.request` is in `TEMPLATE_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):
    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):
        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).

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

```
def submit_plugin_form_data(self, form_entry, request, form):
    # 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(FILES_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 autodiscovered 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):
    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(FILES_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):
    # 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):
    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):  
    uid = "sample_mail"  
    name = _("Sample mail")  
    form = SampleMailForm
```

```
def run(self, form_entry, request, form):
    send_mail(
        self.data.subject,
        json.dumps(form.cleaned_data),
        self.data.from_email,
        [self.data.to_email],
        fail_silently = True
    )
```

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 autodiscovered 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 ugettext_lazy as _
from fobi.base import BasePluginForm
```

Defining the form for Sample mail handler plugin.

```
class MailForm(forms.Form, BasePluginForm):
    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 enties.

    :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):
    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 ugettext_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.core.urlresolvers import reverse
from django.contrib import messages
from django.utils.translation import ugettext_lazy as _

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

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

Defining the wizard view for Sample importer plugin.

```
class SampleImporterWizardView(SessionWizardView):
    """
    """
    form_list = [SampleImporterStep1Form, SampleImporterStep2Form]

    def get_form_kwargs(self, step):
        """
        """
        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):
        # 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(
            '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):  
    stage = CALLBACK_FORM_VALID  
  
    def callback(self, form_entry, request, form):  
        print("Great! Your form is valid!")  
  
form_callback_registry.register(SampleFooCallback)
```

Add the callback module to INSTALLED_APPS.

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

Suggestions

13.1 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.

13.2 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.

14.1 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
```

```
|         -- 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 ugettext_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'
    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-registraton`.
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',
        '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
-- *** 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.

14.2 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.

14.2.1 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
```

14.2.2 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):
    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)
```

14.2.3 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 %}
```

14.2.4 templates/override_simple_theme/snippets/form_ajax.html

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

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

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.

18.1 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.

18.1.1 Fields

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

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

- [Select multiple \(drop-down\)](#)
- [Slug*](#)
- [Select multiple model objects \(drop-down\)](#)
- [Text*](#)
- [Textarea*](#)
- [Time](#)
- [URL*](#)

18.1.2 Content/presentation

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

- [Content image](#): Insert an image.
- [Content text](#): Add text.
- [Content video](#): Add an embed YouTube or Vimeo video.

18.1.3 Security

- [CAPTCHA](#): Captcha integration, requires `django-simple-captcha` package.
- [ReCAPTCHA](#): Captcha integration, requires `django-recaptcha` package.
- [Honeypot](#): Anti-spam honeypot field.

18.1.4 MPTT fields

- [Select MPTT model object \(drop-down\)](#)
- [Select multiple MPTT model objects \(drop-down\)](#)

18.1.5 Test

Test plugins are made for dev purposes only.

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

18.2 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.

18.3 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.

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.

```
TEMPLATE_CONTEXT_PROCESSORS = (
    # ...
    "fobi.context_processors.dynamic_values",
    # ...
)
```

```
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.
 - 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 (unparsed) 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).

23.1 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 %}
```

23.2 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 are two scenarios to deal with missing plugin errors, which you have don't yet have full control of. 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.

Available translations

English is the primary language.

- [Dutch](#) (core and plugins)
- [German](#) (core and plugins)
- [Russian](#) (core and plugins)

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- or 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`.

Troubleshooting

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

License

GPL 2.0/LGPL 2.1

Support

For any issues contact me at the e-mail given in the *Author* section.

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

Screenshots

31.1 Bootstrap3 theme

31.1.1 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](#)

31.1.2 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

31.1.3 View/edit form

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

© 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](#)

Choose form element to add ▾

- 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

© 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*

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

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*	From name
From email*	from@example.com
To name*	To name
To email*	to@example.com
Subject*	My form #1 mail
Body	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.

Add

© 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

© django-fobi example site 2014

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

© django-fobi example site 2014

[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*	Delusional Insanity
URL*	http://youtu.be/8GVlui0JK0M?list=UURKM8j-UZfo8FYY9S3GsfWw
Size	500x400

© django-fobi example site 2014

Build your forms

Add "Boolean" element to the form

Fields marked with * are required

Label*	Agree?
Name*	agree
Help text	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.
Initial	☐
Required	☐

© django-fobi example site 2014

Build your formsEditView

Elements ordering edited successfully.

Edit form

ElementsHandlersProperties

Add elements to your form

Fields marked with * are requiredChoose form element to add

(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 lacus, a commodo lacus tincidunt vitae. Cras accumsan; lacus ut mollis luctus sed.

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

31.1. Bootstrap3 theme

Save ordering

91

© django-fobi example site 2014

Build your forms Edit View

View form

Fields marked with * are required



Name*

Description

Website



Image

No file selected.

Agree?

☐

31.2 Simple theme

31.2.1 View/edit form

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

Home > Fobi > My test form

Edit form

Elements
Handlers
Properties

Change form elements

Add form element +

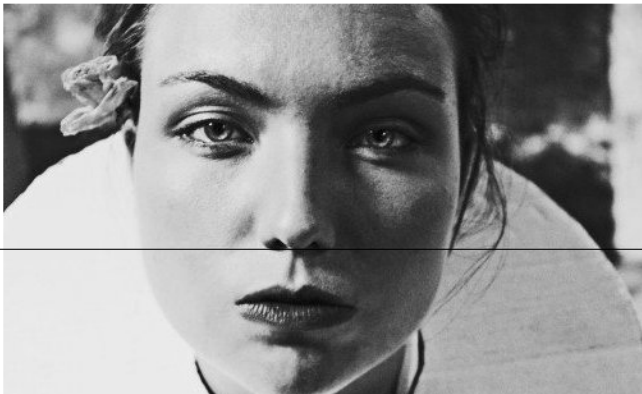
(Content video):
Delusional Insanity - To Far Beyond

Edit Delete

Name (Text):
Edit Delete

Description (Textarea):
Edit Delete

Website (URL):
Edit Delete

(Content image):


Edit form

Elements

Handlers

Properties

Change form elements

(Content video):

Delusional Insanity - To Far Beyond



Edit

Delete

Name (Text):

Edit

Delete

Description (Textarea):

Edit

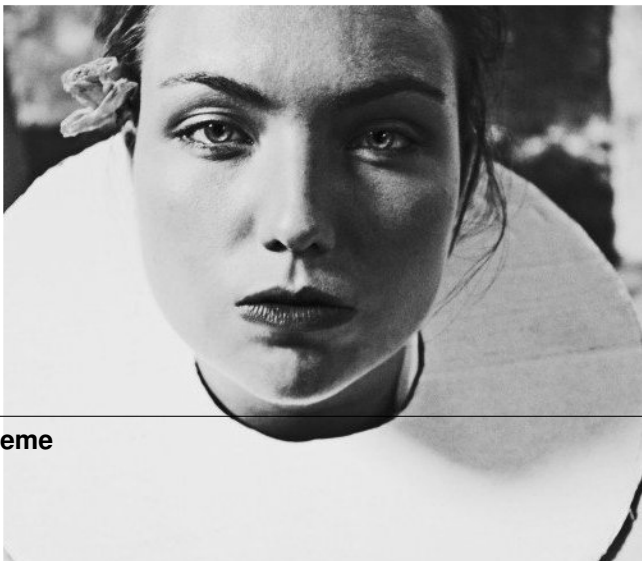
Delete

Website (URL):

Edit

Delete

(Content image):



Add form element +

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

31.2. Simple theme

95

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

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

Home > Fobi > My test form

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 fells, 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.

Contents:

32.1 fobi package

32.1.1 Subpackages

fobi.contrib package

Subpackages

fobi.contrib.apps package

Subpackages

fobi.contrib.apps.djangocms_integration package

Submodules

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'
```

fobi.contrib.apps.djangocms_integration.cms_plugins module

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.

fobi.contrib.apps.djangocms_integration.defaults module

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

fobi.contrib.apps.djangocms_integration.models module

fobi.contrib.apps.djangocms_integration.settings module

- `WIDGET_FORM_SENT` (str): Name of the GET param indicating that form has been successfully sent.

Module contents

fobi.contrib.apps.feincms_integration package

Submodules

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'`

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.

fobi.contrib.apps.feincms_integration.defaults module

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

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.

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

Meta class.

abstract = False

app_label = 'fobi'

`FobiFormWidget.can_redirect` = True

`FobiFormWidget.finalize(request, response)`

Finalize.

`FobiFormWidget.form_entry`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`FobiFormWidget.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.

`FobiFormWidget.form_sent_get_param = 'sent'`

`FobiFormWidget.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.

`FobiFormWidget.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.

`FobiFormWidget.form_title`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FobiFormWidget.get_form_template_name_display(*moreargs, **morekwargs)`

`FobiFormWidget.get_success_page_template_name_display(*moreargs, **morekwargs)`

`FobiFormWidget.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.

`FobiFormWidget.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.

`FobiFormWidget.process(request, **kwargs)`

This is where most of the form handling happens.

Parameters `request` (`django.http.HttpRequest`) –

Return `django.http.HttpResponse` | `str`

`FobiFormWidget.render(**kwargs)`

Render.

`FobiFormWidget.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.

`FobiFormWidget.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.

`FobiFormWidget.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.

Module contents

fobi.contrib.apps.mezzanine_integration package

Submodules

fobi.contrib.apps.mezzanine_integration.admin module

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'

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.

fobi.contrib.apps.mezzanine_integration.defaults module

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

fobi.contrib.apps.mezzanine_integration.models module

fobi.contrib.apps.mezzanine_integration.page_processors module

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.

Module contents

fobi.contrib.apps.wagtail_integration package

Submodules

fobi.contrib.apps.wagtail_integration.apps module

fobi.contrib.apps.wagtail_integration.conf module

fobi.contrib.apps.wagtail_integration.defaults module

fobi.contrib.apps.wagtail_integration.helpers module

fobi.contrib.apps.wagtail_integration.models module

fobi.contrib.apps.wagtail_integration.settings module

Module contents

Module contents

fobi.contrib.plugins package

Subpackages

fobi.contrib.plugins.form_elements package

Subpackages

fobi.contrib.plugins.form_elements.content package

Subpackages

fobi.contrib.plugins.form_elements.content.content_image package

Submodules

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'

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.

fobi.contrib.plugins.form_elements.content.content_image.defaults module

fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements module

class fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements.**ContentImageFormElementPlugin**

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 ContentImageForm

get_form_field_instances (*request*=None)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

post_processor ()

Post process data.

Always the same.

```
uid = 'content_image'
```

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=1, pre-fix=None, initial=None, error_class=None, 'django.forms.la-bel_suffix', empty_perfield_ordering, use_required_if_
```

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

Form for ContentImagePlugin.

```
base_fields = OrderedDict([('file', <django.forms.fields.ImageField object at 0x7f156a8707d0>), ('alt', <django.forms
```

```
declared_fields = OrderedDict([('file', <django.forms.fields.ImageField object at 0x7f156a8707d0>), ('alt', <django.f
```

```
media
```

```
plugin_data_fields = [('file', ''), ('alt', ''), ('fit_method', 'center'), ('size', '500x500')]
```

```
save_plugin_data (request=None)
```

Saving the plugin data and moving the file.

fobi.contrib.plugins.form_elements.content.content_image.helpers module

```
fobi.contrib.plugins.form_elements.content.content_image.helpers.handle_uploaded_file (image_file)
Handle uploaded file.
```

```
Parameters image_file (django.core.files.uploadedfile.InMemoryUploadedFile)
```

```
-
```

```
Return string Path to the image (relative).
```

```
fobi.contrib.plugins.form_elements.content.content_image.helpers.get_crop_filter (fit_method)
Get crop filter.
```

```
fobi.contrib.plugins.form_elements.content.content_image.helpers.delete_file (image_file)
Delete file from disc.
```

```
fobi.contrib.plugins.form_elements.content.content_image.helpers.ensure_unique_filename (destination)
Makes sure filenames are never overwritten.
```

```
Parameters destination (string)-
```

```
Return string
```

```
fobi.contrib.plugins.form_elements.content.content_image.helpers.clone_file (source_filename, relative_path, active_path=True)
Clone the file.
```

Parameters `source_filename` (*string*) – Source filename.

Return string Filename of the cloned file.

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*)

Module contents

fobi.contrib.plugins.form_elements.content.content_text package

Submodules

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'

fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextForm`

Bases: `fobi.base.FormElementPlugin`

Content text plugin.

form

alias of `ContentTextForm`

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

```

post_processor()
    Post process data.

    Always the same.

uid = 'content_text'

```

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=u'ia
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<
'django.form
la-
bel_suffix=N
empty_permit
field_order=
use_required

```

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

Form for ContentTextPlugin.

```
base_fields = OrderedDict([('text', <django.forms.fields.CharField object at 0x7f156a870750>)])
```

```
declared_fields = OrderedDict([('text', <django.forms.fields.CharField object at 0x7f156a870750>)])
```

```
media
```

```
plugin_data_fields = [('text', '')]
```

Module contents

fobi.contrib.plugins.form_elements.content.content_video package

Submodules

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'
```

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.

fobi.contrib.plugins.form_elements.content.content_video.defaults module

fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.ContentVideoForm`

Bases: `fobi.base.FormElementPlugin`

Content video plugin.

form

alias of `ContentVideoForm`

get_form_field_instances (*request=None*)

Get form field instances.

group = `<django.utils.functional.__proxy__ object>`

name = `<django.utils.functional.__proxy__ object>`

post_processor ()

Process plugin data.

Always the same.

uid = `'content_video'`

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=1, pre-fix=None, initial=None, error_class='django.forms.label_suffix', empty_permitted=False, field_order=None, use_required_attribute=True*)

Bases: `django.forms.forms.Form`, `fobi.base.BasePluginForm`

Form for `ContentVideoPlugin`.

base_fields = `OrderedDict`([(`'title'`, <django.forms.fields.CharField object at 0x7f156a8704d0>), (`'url'`, <django.forms

declared_fields = `OrderedDict`([(`'title'`, <django.forms.fields.CharField object at 0x7f156a8704d0>), (`'url'`, <django.

media

plugin_data_fields = [(`'title'`, `'`), (`'url'`, `'`), (`'size'`, `'500x400'`)]

`fobi.contrib.plugins.form_elements.content.content_video.settings` module

Module contents

Module contents

`fobi.contrib.plugins.form_elements.fields` package

Subpackages

`fobi.contrib.plugins.form_elements.fields.boolean` package

Submodules

`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'`

`fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements` module

class `fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.BooleanSelectPlugi`

Bases: `fobi.base.FormFieldPlugin`

Boolean select plugin.

form

alias of `BooleanSelectForm`

get_form_field_instances(*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = `'boolean'`

fobi.contrib.plugins.form_elements.fields.boolean.forms module

Module contents

fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple package

Submodules

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'
```

fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf module

```
fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf.get_setting(setting,
                                                                                   over-
                                                                                   ride=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.
```

fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.defaults module

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.C
    Bases: fobi.base.FormFieldPlugin
    Checkbox select multiple field plugin.

    form
        alias of CheckboxSelectMultipleInputForm

    get_form_field_instances(request=None)
        Get form field instances.

    group = <django.utils.functional.__proxy__ object>
    name = <django.utils.functional.__proxy__ object>
```

submit_plugin_form_data (*form_entry, request, form*)

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'

fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms module

class *fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleForm*

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for CheckboxSelectMultipleInputPlugin.

base_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a7f4a10>), ('name', <django.forms.fields.CharField object at 0x7f156a7f4a10>)])

clean_initial ()

Validating the initial value.

declared_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a7f4a10>), ('name', <django.forms.fields.CharField object at 0x7f156a7f4a10>)])

media

plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''), ('initial', ''), ('required', False)]

fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.date package

Submodules

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'

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 DateInputForm

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry*, *request*, *form*)

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'

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=u'id_%s'*, *pre-fix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *field_order=None*, *use_required_attribute=None*)

Bases: django.forms.forms.Form, *fobi.base.BaseFormFieldPluginForm*

Form for DateInputPlugin.

```

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7f44d0>), ('name', <django.for
clean_initial()
    Clean the initial value.
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7f44d0>), ('name', <djan
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('input_formats', ''), ('required', False)]

```

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'

```

Module contents

fobi.contrib.plugins.form_elements.fields.date_drop_down package

Submodules

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'

```

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.DateDropDown
    Bases: fobi.base.FormFieldPlugin
    Date drop down field plugin.
    form
        alias of DateDropDownInputForm
    get_form_field_instances(request=None)
        Get form field instances.
    group = <django.utils.functional.__proxy__ object>
    name = <django.utils.functional.__proxy__ object>
    uid = 'date_drop_down'

```

fobi.contrib.plugins.form_elements.fields.date_drop_down.forms module

class fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.**DateDropDownInputForm** (da

Bases: django.forms.forms.Form, *fobi.base.BaseFormFieldPluginForm*

Form for DateDropDownInputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7e8d90>), ('name', <django.for

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7e8d90>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('year_min', ''), ('year_max', ''), ('initial', ''), ('input

Module contents

fobi.contrib.plugins.form_elements.fields.datetime package

Submodules

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'

fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.**DateTimeInputPlug**

Bases: *fobi.base.FormFieldPlugin*

DateTime field plugin.

form

alias of DateTimeInputForm

get_form_field_instances (*request=None*)

Get form field instances.

```
group = <django.utils.functional.__proxy__ object>
name = <django.utils.functional.__proxy__ object>
submit_plugin_form_data (form_entry, request, form)
    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'
```

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=u'id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class 'django.forms.util.ErrorList'>,
'django.forms.util.ErrorList',
la-
bel_suffix=None,
empty_permitted=True,
field_order=None,
use_required_attribute=True)
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for DateTimeInputPlugin.

```
base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7e8890>), ('name', <django.forms.fields.CharField object at 0x7f156a7e8890>)])
```

```
clean_initial ()
    Clean the initial value.
```

```
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7e8890>), ('name', <django.forms.fields.CharField object at 0x7f156a7e8890>)])
```

```
media
```

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('input_formats', ''), ('required', False)]
```

fobi.contrib.plugins.form_elements.fields.datetime.widgets module

```
class fobi.contrib.plugins.form_elements.fields.datetime.widgets.BaseDateTimePluginWidget (plugin_uid)
    Bases: fobi.base.FormElementPluginWidget
```

Base datetime form element plugin widget.

```
plugin_uid = 'datetime'
```

Module contents

fobi.contrib.plugins.form_elements.fields.decimal package

Submodules

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'
```

fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.DecimalInputPlugin

    Bases: fobi.base.FormFieldPlugin

    Decimal input plugin.

    form
        alias of DecimalInputForm

    get_form_field_instances(request=None)
        Get form field instances.

    group = <django.utils.functional.__proxy__ object>

    name = <django.utils.functional.__proxy__ object>

    uid = 'decimal'
```

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=u'id_%s',
                                                                                pre-
                                                                                fix=None,
                                                                                ini-
                                                                                tial=None,
                                                                                er-
                                                                                ror_class=<class
                                                                                'django.forms.utils.L
                                                                                la-
                                                                                bel_suffix=None,
                                                                                empty_permitted=Fa
                                                                                field_order=None,
                                                                                use_required_attribu

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm

    Form for DecimalInputPlugin.

    base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7dde90>), ('name', <django.for

    declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7dde90>), ('name', <djan

    media
```

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('max_digits', ''), ('decimal_places', ''),
```

Module contents

fobi.contrib.plugins.form_elements.fields.email package

Submodules

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'
```

fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements.EmailInputPlugin(user)
    Bases: fobi.base.FormFieldPlugin
    Email input plugin.
    form
        alias of EmailInputForm
    get_form_field_instances(request=None)
        Get form field instances.
    group = <django.utils.functional.__proxy__ object>
    name = <django.utils.functional.__proxy__ object>
    uid = 'email'
```

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=u'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=N
                                                                )
    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
    Form for EmailInputPlugin.
```

```
base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7dd810>), ('name', <django.f
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7dd810>), ('name', <djar
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('max_length', '255'), ('required', False
```

Module contents

fobi.contrib.plugins.form_elements.fields.file package

Submodules

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'
```

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.
```

fobi.contrib.plugins.form_elements.fields.file.defaults module

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 FileInputForm
    get_form_field_instances(request=None)
        Get form field instances.
    group = <django.utils.functional.__proxy__ object>
```

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry, request, form*)

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*) –

uid = 'file'

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=u'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
```

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for FileInputPlugin.

base_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a7dd210>), ('name', <django.f

declared_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a7dd210>), ('name', <djar

media

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('max_length', '255'), ('required', False

fobi.contrib.plugins.form_elements.fields.file.settings module

- **FILES_UPLOAD_DIR** (string)

Module contents

fobi.contrib.plugins.form_elements.fields.float package

Submodules

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'

fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements.**FloatInputPlugin** (*user*)

Bases: *fobi.base.FormFieldPlugin*

Float input plugin.

form

alias of FloatInputForm

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = 'float'

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=u'id_%s'*, *pre_fix=None*, *initial=None*, *error_class=<class 'django.forms.utils.ErrorList'>*, *label_suffix=None*, *empty_permitted=False*, *field_order=None*, *use_required_attribute=None*)

Bases: django.forms.forms.Form, *fobi.base.BaseFormFieldPluginForm*

Form for FloatInputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('name', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('help_text', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('initial', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('min_value', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('max_value', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('step', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('required', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('error_messages', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('label_suffix', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('empty_permitted', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('field_order', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('use_required_attribute', <django.forms.fields.CharField object at 0x7f156a7d2b10>)])

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('name', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('help_text', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('initial', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('min_value', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('max_value', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('step', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('required', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('error_messages', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('label_suffix', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('empty_permitted', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('field_order', <django.forms.fields.CharField object at 0x7f156a7d2b10>), ('use_required_attribute', <django.forms.fields.CharField object at 0x7f156a7d2b10>)])

media

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('min_value', None), ('max_value', None), ('step', None), ('required', None), ('error_messages', None), ('label_suffix', None), ('empty_permitted', None), ('field_order', None), ('use_required_attribute', None)]

Module contents

fobi.contrib.plugins.form_elements.fields.hidden package

Submodules

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'
```

fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements.HiddenInputPlugin(u
    Bases: fobi.base.FormFieldPlugin

    Hidden field plugin.

    form
        alias of HiddenInputForm

    get_form_field_instances(request=None)
        Get form field instances.

    group = <django.utils.functional.__proxy__ object>

    is_hidden = True

    name = <django.utils.functional.__proxy__ object>

    uid = 'hidden'
```

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=u'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)

    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm

    Form for HiddenInputPlugin.

    base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7d2710>), ('name', <django.forms.fields.CharField object at 0x7f156a7d2710>)])

    declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7d2710>), ('name', <django.forms.fields.CharField object at 0x7f156a7d2710>)])
```

```
media
```

```
plugin_data_fields = [('label', ''), ('name', ''), ('initial', ''), ('max_length', '255'), ('required', False)]
```

Module contents

fobi.contrib.plugins.form_elements.fields.hidden_model_object package

Submodules

fobi.contrib.plugins.form_elements.fields.hidden_model_object.apps module

fobi.contrib.plugins.form_elements.fields.hidden_model_object.conf module

fobi.contrib.plugins.form_elements.fields.hidden_model_object.defaults module

fobi.contrib.plugins.form_elements.fields.hidden_model_object.fobi_form_elements module

fobi.contrib.plugins.form_elements.fields.hidden_model_object.forms module

fobi.contrib.plugins.form_elements.fields.hidden_model_object.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.input package

Submodules

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'
```

fobi.contrib.plugins.form_elements.fields.input.constants module

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 InputForm

get_form_field_instances (*request=None*)
 Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = 'input'

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=u'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)

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for InputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7c9590>), ('name', <django.for

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7c9590>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('max_length', '255'), ('required', False

Module contents

fobi.contrib.plugins.form_elements.fields.integer package

Submodules

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'
```

fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.**IntegerInputPlugin**

Bases: *fobi.base.FormFieldPlugin*

Integer input plugin.

form

alias of IntegerInputForm

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = 'integer'

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=u'id_%s',

pre-

fix=None,

ini-

tial=None,

er-

ror_class=<class

'django.forms.utils.L

la-

bel_suffix=None,

empty_permitted=Fe

field_order=None,

use_required_attri

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for IntegerInputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7bde50>), ('name', <django.for

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a7bde50>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('min_value', None), ('max_value', Non

Module contents

fobi.contrib.plugins.form_elements.fields.ip_address package

Submodules

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'

fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.**IPAddressInputF**

Bases: *fobi.base.FormFieldPlugin*

IP address field plugin.

form

alias of IPAddressInputForm

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = 'ip_address'

fobi.contrib.plugins.form_elements.fields.ip_address.forms module

Module contents

fobi.contrib.plugins.form_elements.fields.null_boolean package

Submodules

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'

fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.**NullBooleanSe**

Bases: *fobi.base.FormFieldPlugin*

Null boolean select plugin.

form

alias of NullBooleanSelectForm

```

get_form_field_instances (request=None)
    Get form field instances.

group = <django.utils.functional.__proxy__ object>
name = <django.utils.functional.__proxy__ object>
uid = 'null_boolean'

```

fobi.contrib.plugins.form_elements.fields.null_boolean.forms module

Module contents

fobi.contrib.plugins.form_elements.fields.password package

Submodules

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'

```

fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements module

```

class fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements.PasswordInputPlug
    Bases: fobi.base.FormFieldPlugin
    Password field plugin.

    form
        alias of PasswordInputForm

    get_form_field_instances (request=None)
        Get form field instances.

    group = <django.utils.functional.__proxy__ object>
    name = <django.utils.functional.__proxy__ object>
    uid = 'password'

```

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=u'id_%s',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<class
'django.forms.uti
la-
bel_suffix=None,
empty_permitted=
field_order=None
use_required_attr
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for PasswordInputPlugin.

base_fields = `OrderedDict`([(`'label'`, <django.forms.fields.CharField object at 0x7f156a832890>), (`'name'`, <django.for

declared_fields = `OrderedDict`([(`'label'`, <django.forms.fields.CharField object at 0x7f156a832890>), (`'name'`, <djan

media

plugin_data_fields = [(`'label'`, `'`), (`'name'`, `'`), (`'help_text'`, `'`), (`'initial'`, `'`), (`'max_length'`, `'255'`), (`'required'`, `False`)]

Module contents

`fobi.contrib.plugins.form_elements.fields.radio` package

Submodules

`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'`

`fobi.contrib.plugins.form_elements.fields.radio.conf` module

```
fobi.contrib.plugins.form_elements.fields.radio.conf.get_setting (setting, over-
ride=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.

fobi.contrib.plugins.form_elements.fields.radio.defaults module

fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements.**RadioInputPlugin** (*user*)

Bases: *fobi.base.FormFieldPlugin*

Radio field plugin.

form

alias of RadioInputForm

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry, request, form*)

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'

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=u'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=N

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for RadioInputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a832210>), ('name', <django.for

clean_initial ()

Validating the initial value.

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a832210>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''), ('initial', ''), ('required', False)]

fobi.contrib.plugins.form_elements.fields.radio.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.regex package

Submodules

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'

fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.**RegexInputPlugin**(*user*)

Bases: *fobi.base.FormFieldPlugin*

Regex field plugin.

form

alias of RegexInputForm

get_form_field_instances(*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = 'regex'

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=u'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=N
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for RegexInputPlugin.

```
base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a826950>), ('name', <django.for
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a826950>), ('name', <djan
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('regex', ''), ('max_length', '255'), ('req
```

Module contents

`fobi.contrib.plugins.form_elements.fields.select` package

Submodules

`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'
```

`fobi.contrib.plugins.form_elements.fields.select.conf` module

```
fobi.contrib.plugins.form_elements.fields.select.conf.get_setting (setting,
                                                                    over-
                                                                    ride=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.

fobi.contrib.plugins.form_elements.fields.select.defaults module

fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.**SelectInputPlugin** (*u*

Bases: *fobi.base.FormFieldPlugin*

Select field plugin.

form

alias of *SelectInputForm*

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry, request, form*)

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'

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=u'id_%s', pre-fix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList', label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=*

Bases: *django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm*

Form for *SelectInputPlugin*.

base_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a826350>), ('name', <django.for

clean_initial ()

Validating the initial value.

```

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a826350>), ('name', <django.forms.fields.CharField object at 0x7f156a826350>), ('media', <django.forms.fields.CharField object at 0x7f156a826350>), ('plugin_data_fields', <django.forms.fields.CharField object at 0x7f156a826350>)]
media
plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''), ('initial', ''), ('required', False)]

```

fobi.contrib.plugins.form_elements.fields.select.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.select_model_object package

Submodules

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'

```

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.

fobi.contrib.plugins.form_elements.fields.select_model_object.defaults module

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.SelectModelObjectFieldPlugin
    Bases: fobi.base.ModelFormPlugin
    Select model object field plugin.
    form
        alias of SelectModelObjectInputForm

```

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry, request, form*)

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'

fobi.contrib.plugins.form_elements.fields.select_model_object.forms module

class *fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectIn*

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for SelectModelObjectPlugin.

base_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a81ae10>), ('name', <django.for

declared_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a81ae10>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''), ('initial', ''), ('required', False)]

fobi.contrib.plugins.form_elements.fields.select_model_object.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.select_mptt_model_object package

Submodules

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_modul

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'

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.

fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults module

fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.fobi_form_elements module

fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms module

`class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectPluginForm`

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for SelectMPTTModelObjectPlugin.

base_fields = `OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('name', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('model', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('help_text', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('initial', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('required', <django.forms.fields.BooleanField object at 0x7f156718d2d0>), ('media', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('plugin_data_fields', <django.forms.fields.CharField object at 0x7f156718d2d0>)])`

declared_fields = `OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('name', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('model', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('help_text', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('initial', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('required', <django.forms.fields.BooleanField object at 0x7f156718d2d0>), ('media', <django.forms.fields.CharField object at 0x7f156718d2d0>), ('plugin_data_fields', <django.forms.fields.CharField object at 0x7f156718d2d0>)])`

media

plugin_data_fields = `[('label', ''), ('name', ''), ('model', ''), ('help_text', ''), ('initial', ''), ('required', False)]`

fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.select_multiple package

Submodules

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'`

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.

fobi.contrib.plugins.form_elements.fields.select_multiple.defaults module

fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultiple`

Bases: `fobi.base.FormFieldPlugin`

Select multiple field plugin.

form

alias of `SelectMultipleInputForm`

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry, request, form*)

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'

fobi.contrib.plugins.form_elements.fields.select_multiple.forms module

class fobi.contrib.plugins.form_elements.fields.select_multiple.forms.**SelectMultipleInputForm**

Bases: django.forms.forms.Form, *fobi.base.BaseFormFieldPluginForm*

Form for SelectMultipleInputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a81a810>), ('name', <django.for

clean_initial()

Validating the initial value.

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a81a810>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''), ('initial', ''), ('required', False)]

fobi.contrib.plugins.form_elements.fields.select_multiple.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects package

Submodules

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*, *app*)

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'

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** (*se*, *ov*, *ri*)

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.

fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults module

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*

Bases: *fobi.base.FormFieldPlugin*

Select multiple model objects field plugin.

form

alias of *SelectMultipleModelObjectsInputForm*

get_form_field_instances (*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry, request, form*)

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'

fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms module

class *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms.SelectMul*

Bases: *django.forms.forms.Form*, *fobi.base.BaseFormFieldPluginForm*

Form for *SelectMultipleModelObjectsPlugin*.

base_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a80fc10>), ('name', <django.for

declared_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a80fc10>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''), ('initial', ''), ('required', False)]

fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects package

Submodules

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.Confi`

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'

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_sett`

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.

fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults module

fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_elements module

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.Sele`

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for SelectMultipleMPTTModelObjectsPlugin.

base_fields = `OrderedDict([('label', <django.forms.fields.CharField object at 0x7f1567c45250>), ('name', <django.for`

declared_fields = `OrderedDict([('label', <django.forms.fields.CharField object at 0x7f1567c45250>), ('name', <djan`

media

plugin_data_fields = `[('label', ''), ('name', ''), ('model', ''), ('help_text', ''), ('initial', ''), ('required', False)]`

fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.settings module

Module contents

fobi.contrib.plugins.form_elements.fields.slug package

Submodules

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'

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 SlugInputForm

get_form_field_instances(*request=None*)

Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = 'slug'

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=u'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)

Bases: django.forms.forms.Form, *fobi.base.BaseFormFieldPluginForm*

Form for SlugInputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a80f690>), ('name', <django.for

```

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a80f690>), ('name', <djan
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('max_length', '255'), ('required', False

```

Module contents

fobi.contrib.plugins.form_elements.fields.text package

Submodules

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'

```

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 TextInputForm
    get_form_field_instances(request=None)
        Get form field instances.
    group = <django.utils.functional.__proxy__ object>
    name = <django.utils.functional.__proxy__ object>
    uid = 'text'

```

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=u'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

Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
Form for TextInputPlugin.
base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a80f110>), ('name', <django.for
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a80f110>), ('name', <djan
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('max_length', '255'), ('required', False
```

Module contents

fobi.contrib.plugins.form_elements.fields.textarea package

Submodules

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'
```

fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements module

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=u'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=
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for TextareaPlugin.

```
base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a800bd0>), ('name', <django.f
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a800bd0>), ('name', <djar
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('required', False), ('placeholder', '')]
```

Module contents

`fobi.contrib.plugins.form_elements.fields.time` package

Submodules

`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'
```

`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 TimeInputForm

get_form_field_instances (request=None)
    Get form field instances.

group = <django.utils.functional.__proxy__ object>
```

name = <django.utils.functional.__proxy__ object>

submit_plugin_form_data (*form_entry, request, form*)
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'

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=u'id_%s', pre-fix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList', label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=None*)

Bases: *django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm*

Form for TimeInputPlugin.

base_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a8006d0>), ('name', <django.for

clean_initial ()

Clean the initial value.

declared_fields = *OrderedDict*([('label', <django.forms.fields.CharField object at 0x7f156a8006d0>), ('name', <djan

media

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('input_formats', ''), ('required', False)

Module contents

fobi.contrib.plugins.form_elements.fields.url package

Submodules

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'
```

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 URLInputForm
    get_form_field_instances(request=None)
        Get form field instances.
    group = <django.utils.functional.__proxy__ object>
    name = <django.utils.functional.__proxy__ object>
    uid = 'url'
```

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=u'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)
    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
    Form for URLPlugin.
    base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a800150>), ('name', <django.for
    declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a800150>), ('name', <djan
    media
    plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('max_length', '255'), ('required', False
```

Module contents

Module contents

fobi.contrib.plugins.form_elements.security package

Subpackages

fobi.contrib.plugins.form_elements.security.captcha package

Submodules

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'
```

fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaInputPlugin
    Bases: fobi.base.FormElementPlugin
    Captcha field plugin.
    form
        alias of CaptchaInputForm
    get_form_field_instances(request=None)
        Get form field instances.
    group = <django.utils.functional.__proxy__ object>
    name = <django.utils.functional.__proxy__ object>
    uid = 'captcha'
```

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=u'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_attr
```

Bases: django.forms.forms.Form, *fobi.base.BaseFormFieldPluginForm*

Form for CaptchaInputPlugin.

```
base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f1567694750>), ('name', <django.for
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f1567694750>), ('name', <djan
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('required', True)]
```

Module contents

fobi.contrib.plugins.form_elements.security.honeypot package

Submodules

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'
```

fobi.contrib.plugins.form_elements.security.honeypot.conf module

```
fobi.contrib.plugins.form_elements.security.honeypot.conf.get_setting(setting,
                                                                    over-
                                                                    ride=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.
```

fobi.contrib.plugins.form_elements.security.honeypot.defaults module

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,
                                                                    *args,
                                                                    **kwargs)
    Bases: django.forms.fields.CharField
    clean(value)
        Check that honeypot value remained the same.
```

```
default_error_messages = {'invalid': <django.utils.functional.__proxy__ object at 0x7f156a8e0110>}
```

widget

alias of `HiddenInput`

fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin
```

Bases: `fobi.base.FormElementPlugin`

Honeypot field plugin.

form

alias of `HoneypotInputForm`

```
get_form_field_instances(request=None)
```

Get form field instances.

```
group = <django.utils.functional.__proxy__ object>
```

```
is_hidden = True
```

```
name = <django.utils.functional.__proxy__ object>
```

```
uid = 'honeypot'
```

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=u'id_
```

```
pre-
```

```
fix=None,
```

```
ini-
```

```
tial=None,
```

```
er-
```

```
ror_class=<cl
```

```
'django.forms.
```

```
la-
```

```
bel_suffix=No
```

```
empty_permit
```

```
field_order=N
```

```
use_required_
```

Bases: `django.forms.forms.Form`, `fobi.base.BaseFormFieldPluginForm`

Form for `HoneypotInputPlugin`.

```
base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a870d10>), ('name', <django.for
```

```
declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f156a870d10>), ('name', <djan
```

media

```
plugin_data_fields = [('label', ''), ('name', ''), ('initial', ''), ('max_length', '255'), ('required', True)]
```

fobi.contrib.plugins.form_elements.security.honeypot.settings module

- `HONEYPOT_VALUE` (string)

Module contents

fobi.contrib.plugins.form_elements.security.recaptcha package

Submodules

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'
```

fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements.ReCaptchaInput

Bases: fobi.base.FormElementPlugin

ReCaptcha field plugin.

form
    alias of ReCaptchaInputForm

get_form_field_instances(request=None)
    Get form field instances.

group = <django.utils.functional.__proxy__ object>

name = <django.utils.functional.__proxy__ object>

uid = 'recaptcha'
```

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='u', pre-fix=None, initial=None, error_class=None, 'django.forms'
la-bel_suffix=None, empty_permitted=None, field_order=None, use_required_attribute=None)

Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm

Form for ReCaptchaInputPlugin.

base_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f1567b50fd0>), ('name', <django.forms.fields.CharField object at 0x7f1567b50fd0>)])

declared_fields = OrderedDict([('label', <django.forms.fields.CharField object at 0x7f1567b50fd0>), ('name', <django.forms.fields.CharField object at 0x7f1567b50fd0>)])

media
```

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('required', True)]
```

Module contents

Module contents

fobi.contrib.plugins.form_elements.test package

Subpackages

fobi.contrib.plugins.form_elements.test.dummy package

Submodules

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'
```

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)  
        Get form field instances.  
    group = <django.utils.functional.__proxy__ object>  
    name = <django.utils.functional.__proxy__ object>  
    post_processor()  
        Post process data.  
        Always the same.  
    uid = 'dummy'
```

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'
```

Module contents

Module contents

Module contents

fobi.contrib.plugins.form_handlers package

Subpackages

fobi.contrib.plugins.form_handlers.db_store package

Submodules

fobi.contrib.plugins.form_handlers.db_store.admin module

class fobi.contrib.plugins.form_handlers.db_store.admin.**SavedFormDataEntryAdmin** (*model, admin_site*)

Bases: django.contrib.admin.options.ModelAdmin

Saved form data entry admin.

class Media

Media class.

js = ('/static/js/jquery-1.10.2.min.js', '/static/db_store/js/db_store.js', '/static/db_store/js/jquery.expander.min.js')

class SavedFormDataEntryAdmin.**Meta**

Meta class.

app_label = <django.utils.functional.__proxy__ object>

SavedFormDataEntryAdmin.**actions** = ['export_data']

SavedFormDataEntryAdmin.**export_data** (*request, queryset*)

Export data into XLS.

SavedFormDataEntryAdmin.**fieldsets** = ((None, {'fields': ('form_entry', 'user')}), (<django.utils.functional.__proxy__ object>))

SavedFormDataEntryAdmin.**get_queryset** (*request*)

Get queryset.

Used internally.

SavedFormDataEntryAdmin.**list_display** = ('form_entry', 'user', 'formatted_saved_data', 'created')

SavedFormDataEntryAdmin.**list_filter** = ('form_entry', 'user')

SavedFormDataEntryAdmin.**media**

SavedFormDataEntryAdmin.**readonly_fields** = ('created', 'formatted_saved_data')

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'`

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.

fobi.contrib.plugins.form_handlers.db_store.defaults module

fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers module

`class fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreHandlerPlugin` (user

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 = <django.utils.functional.__proxy__ object>

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 = 'db_store'

fobi.contrib.plugins.form_handlers.db_store.helpers module

class fobi.contrib.plugins.form_handlers.db_store.helpers.**DataExporter** (*queryset*)

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.

fobi.contrib.plugins.form_handlers.db_store.models module

class fobi.contrib.plugins.form_handlers.db_store.models.**SavedFormDataEntry** (**args, **kwargs*)

Bases: django.db.models.base.Model

Saved form data.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception SavedFormDataEntry.MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

SavedFormDataEntry.**created**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

SavedFormDataEntry.**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.

SavedFormDataEntry.**form_entry**

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

SavedFormDataEntry.**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.

SavedFormDataEntry.**formatted_saved_data** ()

Shows the formatted saved data records.

Return string

SavedFormDataEntry.**get_next_by_created** (**moreargs, **morekwargs*)

SavedFormDataEntry.**get_previous_by_created** (**moreargs, **morekwargs*)

SavedFormDataEntry.**id**

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`SavedFormDataEntry.objects` = <django.db.models.manager.Manager object>

`SavedFormDataEntry.saved_data`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`SavedFormDataEntry.user`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`SavedFormDataEntry.user_id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

fobi.contrib.plugins.form_handlers.db_store.settings module

- `CSV_DELIMITER` (string)
- `CSV_QUOTECHAR` (string)

fobi.contrib.plugins.form_handlers.db_store.urls module

fobi.contrib.plugins.form_handlers.db_store.views module

`fobi.contrib.plugins.form_handlers.db_store.views.view_saved_form_data_entries` (*request*, *args, **kwargs)

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.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'

Module contents

fobi.contrib.plugins.form_handlers.http_repost package

Submodules

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'

fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers module

class `fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostHandlerPlugin`

Bases: `fobi.base.FormHandlerPlugin`

HTTP repost handler plugin.

Makes a HTTP repost to a given endpoint. Should be executed before `db_store` plugin.

form
alias of `HTTPRepostForm`

name = <django.utils.functional.__proxy__ object>

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'

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=u'id_%s',
                                                                    pre-
                                                                    fix=None,
                                                                    ini-
                                                                    tial=None,
                                                                    er-
                                                                    ror_class=<class
                                                                    'django.forms.utils.ErrorLi
                                                                    la-
                                                                    bel_suffix=None,
                                                                    empty_permitted=False,
                                                                    field_order=None,
                                                                    use_required_attribute=Non

Bases: django.forms.forms.Form, fobi.base.BasePluginForm
Form for HTTPRepostPlugin.
base_fields = OrderedDict([('endpoint_url', <django.forms.fields.URLField object at 0x7f156a870310>)])
declared_fields = OrderedDict([('endpoint_url', <django.forms.fields.URLField object at 0x7f156a870310>)])
media
plugin_data_fields = [('endpoint_url', '')]
```

Module contents

fobi.contrib.plugins.form_handlers.mail package

Submodules

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'

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.

fobi.contrib.plugins.form_handlers.mail.defaults module

fobi.contrib.plugins.form_handlers.mail.fields module

```
class fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField(required=True,
                                                                    wid-
                                                                    get=None,
                                                                    la-
                                                                    bel=None,
                                                                    ini-
                                                                    tial=None,
                                                                    help_text=u'',
                                                                    er-
                                                                    ror_messages=None,
                                                                    show_hidden_initial=False,
                                                                    valida-
                                                                    tors=[],
                                                                    local-
                                                                    ize=False,
                                                                    dis-
                                                                    abled=False,
                                                                    la-
                                                                    bel_suffix=None)
```

Bases: `django.forms.fields.Field`

MultiEmailField.

code = 'invalid'

message = <django.utils.functional.__proxy__ object>

to_python (*value*)

Normalize data to a list of strings.

validate (*value*)

Check if value consists only of valid emails.

widget

alias of MultiEmailWidget

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`

Mail handler plugin.

Sends emails to the person specified. Should be executed before `db_store` and `http_repost` plugins.

form

alias of MailForm

name = <django.utils.functional.__proxy__ object>

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'

fobi.contrib.plugins.form_handlers.mail.forms module

```
class fobi.contrib.plugins.form_handlers.mail.forms.MailForm(data=None,
                                                            files=None,
                                                            auto_id=u'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)
```

Bases: *django.forms.forms.Form*, *fobi.base.BasePluginForm*

Form for BooleanSelectPlugin.

base_fields = OrderedDict([('from_name', <django.forms.fields.CharField object at 0x7f156a859e10>), ('from_email',

declared_fields = OrderedDict([('from_name', <django.forms.fields.CharField object at 0x7f156a859e10>), ('from_

media

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

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.

fobi.contrib.plugins.form_handlers.mail.settings module

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

    media

    prep_value (value)
        Prepare value before effectively render widget

    render (name, value, attrs=None)
        Render.
```

Module contents

Module contents

fobi.contrib.plugins.form_importers package

Subpackages

fobi.contrib.plugins.form_importers.mailchimp_importer package

Submodules

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'
```

fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers module

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImp
    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 = {'initial': 'default', 'name': 'tag', 'required': 'req', 'choices': 'choices', 'help_text': 'help_text'}

field_type_prop_name = 'field_type'

fields_mapping = {'url': 'url', 'radio': 'radio', 'zip': 'text', 'dropdown': 'select', 'date': 'date', 'text': 'text', 'address': 'address'}

name = <django.utils.functional.__proxy__ object>

position_prop_name = 'order'

templates = ['mailchimp_importer/0.html', 'mailchimp_importer/1.html']

uid = 'mailchimp'

wizard

alias of MailchimpImporterWizardView

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=None, pre_fix=None, initial=None, error_class=None, django_label=None, label_sufffix=None, empty_placeholder=None, field_order=None, use_required_attribute=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 = OrderedDict([('api_key', <django.forms.fields.CharField object at 0x7f156a83d850>)])

declared_fields = OrderedDict([('api_key', <django.forms.fields.CharField object at 0x7f156a83d850>)])

media

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 = OrderedDict([('list_id', <django.forms.fields.ChoiceField object at 0x7f156a85cd50>)])

declared_fields = OrderedDict([('list_id', <django.forms.fields.ChoiceField object at 0x7f156a85cd50>)])

media

fobi.contrib.plugins.form_importers.mailchimp_importer.views module

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.views.MailchimpImporterWizardView
    Bases: formtools.wizard.views.SessionWizardView

    MailchimpImporterWizardView.

    done (form_list, **kwargs)

    form_list = [<class 'fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpAPIKeyForm'>, <class
    get_form_kwargs (step)
        Get form kwargs.
```

Module contents

Module contents

Module contents

fobi.contrib.themes package

Subpackages

fobi.contrib.themes.bootstrap3 package

Subpackages

fobi.contrib.themes.bootstrap3.widgets package

Subpackages

fobi.contrib.themes.bootstrap3.widgets.form_elements package

Subpackages

fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget package

Submodules

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
    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'
```

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_e`

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

theme_uid = 'bootstrap3'

Module contents

fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget package

Submodules

fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps module

class `fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps.Co`

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'

fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_form_elements module

class `fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_fo`

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.date

theme_uid = 'bootstrap3'

Module contents

fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget package

Submodules

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`

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'`

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.BaseDummyPluginWidget`

Bases: `fobi.contrib.plugins.form_elements.test.dummy.widgets.BaseDummyPluginWidget`

Dummy plugin widget for Bootstrap 3.

media_css = []

media_js = []

theme_uid = `'bootstrap3'`

Module contents

Module contents

Module contents

Submodules

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'`

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'`

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'
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'
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'
form_properties_snippet_template_name = 'bootstrap3/snippets/form_properties_snippet.html'
form_snippet_template_name = 'bootstrap3/snippets/form_snippet.html'
form_wizard_template = 'bootstrap3/snippets/form_wizard.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/
messages_snippet_template_name = 'bootstrap3/snippets/messages_snippet.html'
name = <django.utils.functional.__proxy__ object>
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'
```

Module contents

fobi.contrib.themes.djangocms_admin_style_theme package

Subpackages

fobi.contrib.themes.djangocms_admin_style_theme.widgets package

Subpackages

fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers package

Subpackages

fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store package

Submodules

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.Con`

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'

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_for`

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 = ''

Module contents

Module contents

Module contents

Submodules

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'
```

fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes module

```
class fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme(us

    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.htm
    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.htm
    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.htm
    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.htm
    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_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.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 = <django.utils.functional.__proxy__ object>
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'

```

Module contents

fobi.contrib.themes.foundation5 package

Subpackages

fobi.contrib.themes.foundation5.widgets package

Subpackages

fobi.contrib.themes.foundation5.widgets.form_elements package

Subpackages

fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget package

Submodules

fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps module

class `fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps.Conf`

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'`

fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form_elements module

class `fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form`

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-four`

theme_uid = `'foundation5'`

Module contents

fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget package

Submodules

fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps module

class `fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps.`

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'`

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`

Bases: `fobi.contrib.plugins.form_elements.fields.datetime.widgets.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.datet`

theme_uid = `'foundation5'`

Module contents

fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget package

Submodules

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**

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'

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.**FormElements**

Bases: *fobi.contrib.plugins.form_elements.test.dummy.widgets.BaseDummyPluginWidget*

Dummy plugin widget for Foundation 5.

media_css = []

media_js = []

theme_uid = 'foundation5'

Module contents

Module contents

fobi.contrib.themes.foundation5.widgets.form_handlers package

Subpackages

fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget package

Submodules

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**

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'

fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_elements module

```
class fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_
    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_entr
```

Module contents

Module contents

Module contents

Submodules

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'
```

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'
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'
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'
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/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')
messages_snippet_template_name = 'foundation5/snippets/messages_snippet.html'
name = <django.utils.functional.__proxy__ object>
uid = 'foundation5'
view_form_entry_ajax_template = 'foundation5/view_form_entry_ajax.html'
view_form_entry_template = 'foundation5/view_form_entry.html'

```

Module contents

fobi.contrib.themes.simple package

Subpackages

fobi.contrib.themes.simple.widgets package

Subpackages

fobi.contrib.themes.simple.widgets.form_handlers package

Subpackages

fobi.contrib.themes.simple.widgets.form_handlers.db_store package

Submodules

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'

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.DbStorePlu
```

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 = ''

Module contents

Module contents

Module contents

Submodules

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'

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'
    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'
    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_snippet_template_name = 'simple/snippets/form_view_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.utils.functional.__proxy__ object>
uid = 'simple'
view_form_entry_ajax_template = 'simple/view_form_entry_ajax.html'
view_form_entry_template = 'simple/view_form_entry.html'

```

Module contents

Module contents

Module contents

fobi.integration package

Submodules

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

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'`

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 = `None`

Module contents

fobi.management package

Subpackages

fobi.management.commands package

Submodules

fobi.management.commands.fobi_find_broken_entries module

class `fobi.management.commands.fobi_find_broken_entries.Command` (*stdout=None, stderr=None, no_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.

fobi.management.commands.fobi_migrate_03_to_04 module

class `fobi.management.commands.fobi_migrate_03_to_04.Command` (*stdout=None, stderr=None, no_color=False*)

Bases: `django.core.management.base.BaseCommand`

Database related changes necessary to upgrade `fobi==0.3.*` to `fobi==0.4`.

The full list of changes is listed below:

- Change the “birthday” occurrences to “date_drop_down”.

handle (**args, **options*)

Handle.

fobi.management.commands.fobi_sync_plugins module

class fobi.management.commands.fobi_sync_plugins.**Command** (*stdout=None, stderr=None, no_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

handle (**args, **options*)
Handle.

fobi.management.commands.fobi_update_plugin_data module

class fobi.management.commands.fobi_update_plugin_data.**Command** (*stdout=None, stderr=None, no_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.

Module contents

Module contents

fobi.templatetags package

Submodules

fobi.templatetags.fobi_tags module

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_fobi_form_handler_plugin_custom_actions** (*parser, to-ken*)

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 [context_var_name] %}
```

Example

```
{% get_fobi_form_handler_plugin_custom_actions plugin form_entry as form_handler_plugin_custom_actions %}
```

`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 %}`

fobi.templatetags.future_compat module

`fobi.templatetags.future_compat.firstof(parser, token, escape=False)`

Outputs the first variable passed that is not False.

Outputs the first variable passed that is not False, without escaping.

Outputs nothing if all the passed variables are False.

Sample usage:

```
{% firstof var1 var2 var3 %}
```

This is equivalent to:

```
{% if var1 %}
    {{ var1|safe }}
{% elif var2 %}
    {{ var2|safe }}
{% elif var3 %}
    {{ var3|safe }}
{% 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 escape the output, use a filter tag:

```
{% filter force_escape %}
    {% firstof var1 var2 var3 "fallback value" %}
{% endfilter %}
```

Module contents

fobi.tests package

Submodules

fobi.tests.base module

`fobi.tests.base.print_info(func)`

Prints some useful info.

`fobi.tests.base.skip(func)`

Simply skips the test.

`fobi.tests.base.is_fobi_setup_completed()`

`fobi.tests.base.mark_fobi_setup_as_completed()`

fobi.tests.constants module

fobi.tests.data module

fobi.tests.helpers module

`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.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.create_form_with_entries(user=None, create_entries_if_form_exist=True)`

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).

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()`

Cleans up the database by removing all form element and form handler entries.

fobi.tests.test_browser_build_dynamic_forms module

`class fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuldDynamicFormsTest (methodName=)`

Bases: `django.test.testcases.LiveServerTestCase`

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

e = `AttributeError`("'Settings' object has no attribute 'LIVE_SERVER_URL'",)

classmethod setUpClass()

Set up class.

tearDown()

Tear down.

classmethod tearDownClass()

Tear down class.

test_1001_open_dashboard (*args, **kwargs)

test_2001_add_form (*args, **kwargs)

test_2002_edit_form (*args, **kwargs)

test_2003_delete_form (*args, **kwargs)

test_2004_submit_form (*args, **kwargs)

test_3001_add_form_elements (*args, **kwargs)

test_3002_remove_form_elements (*args, **kwargs)

test_3003_edit_form_elements (*args, **kwargs)

test_4001_add_form_handlers (*args, **kwargs)

test_4002_remove_form_handlers (*args, **kwargs)

```
test_4003_edit_form_handlers (*args, **kwargs)
```

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)
    test_02_get_registered_form_handler_plugins (*args, **kwargs)
    test_03_get_registered_form_callbacks (*args, **kwargs)
    test_04_get_registered_themes (*args, **kwargs)
    test_05_action_url (*args, **kwargs)
```

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)
```

fobi.tests.test_form_importers_mailchimp module

```
fobi.tests.test_form_importers_mailchimp.do ()
```

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)
    test_02_sortable_dict_move_after_key (*args, **kwargs)
```

Module contents

fobi.urls package

Submodules

fobi.urls.edit module

fobi.urls.view module

Module contents

32.1.2 Submodules

32.1.3 fobi.admin module

```
class fobi.admin.BasePluginModelAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    Base plugin admin.

    class Meta
        Meta.

        app_label = <django.utils.functional.__proxy__ object>

    BasePluginModelAdmin.bulk_change_plugins(*args, **kwargs)
        Bulk change plugins.

        This is where the data is actually processed.

    BasePluginModelAdmin.fieldsets = ((None, {'fields': ('plugin_uid', 'users', 'groups')}),)

    BasePluginModelAdmin.filter_horizontal = ('users', 'groups')

    BasePluginModelAdmin.get_queryset(request)
        Internal method used in get_queryset or queryset methods.

    BasePluginModelAdmin.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.

    BasePluginModelAdmin.list_display = ('plugin_uid_admin', 'users_list', 'groups_list')

    BasePluginModelAdmin.media

    BasePluginModelAdmin.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 at 0x7f156a94f050>]

    get_urls()
        Get URLs.

    media
```

```

class fobi.admin.FormElementEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    FormElementEntry admin.

    class Meta
        Meta.

        app_label = <django.utils.functional.__proxy__ object>

    FormElementEntryAdmin.fieldsets = ((<django.utils.functional.__proxy__ object at 0x7f156a799590>, {'fields': (
    FormElementEntryAdmin.get_queryset(request)
        Internal method used in get_queryset or queryset methods.

    FormElementEntryAdmin.list_display = ('plugin_uid', 'plugin_uid_code', 'plugin_data', 'position', 'form_entry')
    FormElementEntryAdmin.list_editable = ('position',)
    FormElementEntryAdmin.list_filter = ('form_entry', 'plugin_uid')
    FormElementEntryAdmin.media
    FormElementEntryAdmin.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 FormElementEntryForm

    media
    model
        alias of FormElementEntry

class fobi.admin.FormEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin

    FormEntry admin.

    class Meta
        Meta.

        app_label = <django.utils.functional.__proxy__ object>

    FormEntryAdmin.fieldsets = ((<django.utils.functional.__proxy__ object at 0x7f156aa07890>, {'fields': ('name', 'is_public', 'is_cloneable', 'created', 'updated', 'slug', 'user')}),)
    FormEntryAdmin.inlines = [<class 'fobi.admin.FormElementEntryInlineAdmin'>, <class 'fobi.admin.FormHandlerAdmin'>]
    FormEntryAdmin.list_display = ('name', 'slug', 'user', 'is_public', 'created', 'updated', 'is_cloneable')
    FormEntryAdmin.list_editable = ('is_public', 'is_cloneable')
    FormEntryAdmin.list_filter = ('is_public', 'is_cloneable')
    FormEntryAdmin.media
    FormEntryAdmin.radio_fields = {'user': 2}
    FormEntryAdmin.readonly_fields = ('slug',)

```

```

class fobi.admin.FormFieldsetEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    FormEieldsetEntry admin.
    class Meta
        app_label = <django.utils.functional.__proxy__ object>
    FormFieldsetEntryAdmin.fieldsets = ((None, {'fields': ('form_entry', 'name', 'is_repeatable')})),)
    FormFieldsetEntryAdmin.list_display = ('form_entry', 'name', 'is_repeatable')
    FormFieldsetEntryAdmin.list_editable = ('is_repeatable',)
    FormFieldsetEntryAdmin.list_filter = ('is_repeatable',)
    FormFieldsetEntryAdmin.media

class fobi.admin.FormHandlerAdmin(model, admin_site)
    Bases: fobi.admin.BasePluginModelAdmin
    FormHandler admin.
    actions = [<function bulk_change_form_handler_plugins at 0x7f156a94f0c8>]
    get_urls()
        Get URLs.
    media

class fobi.admin.FormHandlerEntryAdmin(model, admin_site)
    Bases: django.contrib.admin.options.ModelAdmin
    FormHandlerEntry admin.
    class Meta
        Meta.
        app_label = <django.utils.functional.__proxy__ object>
    FormHandlerEntryAdmin.fieldsets = ((<django.utils.functional.__proxy__ object at 0x7f156a799790>, {'fields': (
    FormHandlerEntryAdmin.get_queryset(request)
        Internal method used in get_queryset or queryset methods.
    FormHandlerEntryAdmin.list_display = ('plugin_uid', 'plugin_uid_code', 'plugin_data', 'form_entry')
    FormHandlerEntryAdmin.list_filter = ('form_entry', 'plugin_uid')
    FormHandlerEntryAdmin.media
    FormHandlerEntryAdmin.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 FormHandlerEntryForm
    media

```

model

alias of `FormHandlerEntry`

`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.

32.1.4 fobi.app module

`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

`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

32.1.5 fobi.apps module

class `fobi.apps.Config` (*app_name, app_module*)
Bases: `django.apps.config AppConfig`
Config.
label = 'fobi'
name = 'fobi'

32.1.6 fobi.base module

class `fobi.base.BaseDataStorage`
Bases: `object`
Base storage data.
class `fobi.base.FormElementPluginDataStorage`
Bases: `fobi.base.BaseDataStorage`
Storage for *FormField* data.

class `fobi.base.FormHandlerPluginDataStorage`

Bases: `fobi.base.BaseDataStorage`

Storage for *FormField* data.

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.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’, ‘WYSI-WYG’, ‘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 dash 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.ModelForm` Subclass of `django.forms.ModelForm` 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 = <django.utils.functional.__proxy__ object>

help_text = None

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 = []

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, re-define 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.FormHandlerEntry`.

Return dict Should return a dictionary containing data of fields to be updated.

widget = None

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_form_field_instances (*request=None*)
Get the instances of form fields, that plugin contains.

Parameters request (*django.http.HttpRequest*) –

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 *FormElementPluginDataStorage*

submit_plugin_form_data (*form_entry, request, form*)

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*) –

class *fobi.base*.**FormHandlerPlugin** (*user=None*)

Bases: *fobi.base.BasePlugin*

Form handler plugin.

Property *fobi.base*.**FormHandlerPlugin***DataStorage* **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

storage

alias of *FormHandlerPluginDataStorage*

class *fobi.base.FormCallback*

Bases: object

Base 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*) –

stage = None

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 couldn'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*) –

```

        :return mixed.

    plugin_not_found_error_message = "Can't find plugin with uid '{0}' in '{1}' registry."

    plugin_not_found_exception_cls
        alias of DoesNotExist

    register(cls, force=False)
        Registers the plugin in the registry.

        :param mixed.

    type = None

    unregister(cls)
        Un-register.

class fobi.base.FormElementPluginRegistry
    Bases: fobi.base.BaseRegistry

    Form element plugins registry.

    fail_on_missing_plugin = True

    plugin_not_found_exception_cls
        alias of FormElementPluginDoesNotExist

    type = (<class 'fobi.base.FormElementPlugin'>, <class 'fobi.base.FormFieldPlugin'>)

class fobi.base.FormHandlerPluginRegistry
    Bases: fobi.base.BaseRegistry

    Form handler plugins registry.

    fail_on_missing_plugin = True

    plugin_not_found_exception_cls
        alias of FormHandlerPluginDoesNotExist

    type
        alias of FormHandlerPlugin

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.

        :param mixed.

    uidfy(cls)
        Makes a UID string from the class given.

        Return string

```


class fobi.base.**ClassProperty**

Bases: `property`

`ClassProperty`.

fobi.base.**classproperty**

alias of `ClassProperty`

fobi.base.**get_registered_plugins** (*registry*, *as_instances=False*, *sort_items=True*)

Get registered plugins.

Get a list of registered plugins in a form if 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_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_form_element_plugins** ()

Get registered form element plugins.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet autodiscovered, autodiscovers them.

Return list

fobi.base.**get_registered_form_element_plugin_uids** (*flatten=True*)

Get registered form element 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.**validate_form_element_plugin_uid** (*plugin_uid*)

Validate the form element plugin uid.

Parameters *plugin_uid* (*string*) –

Return bool

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_handler_plugin_uids` (*flatten=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.validate_form_handler_plugin_uid` (*plugin_uid*)

Validate the plugin uid.

Parameters `plugin_uid` (*string*) –

Return bool

`fobi.base.get_registered_form_callbacks` (*stage=None*)

Get registered form callbacks for the stage given.

`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.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.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.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.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_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.validate_theme_uid(plugin_uid)`

Validate the theme uid.

Parameters `plugin_uid` (*string*) –

Return bool

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.FormFieldPlugin` (*user=None*)

Bases: `fobi.base.FormElementPlugin`

Form field plugin.

has_value = True

class `fobi.base.FormElementPluginWidgetRegistry`

Bases: `fobi.base.BasePluginWidgetRegistry`

Registry of form element plugins.

type

alias of `FormElementPluginWidget`

class `fobi.base.FormHandlerPluginWidgetRegistry`

Bases: `fobi.base.BasePluginWidgetRegistry`

Registry of form handler plugins.

type

alias of `FormHandlerPluginWidget`

class `fobi.base.FormElementPluginWidget(plugin)`

Bases: `fobi.base.BasePluginWidget`

Form element plugin widget.

storage

alias of `FormElementPluginWidgetDataStorage`

class `fobi.base.FormHandlerPluginWidget(plugin)`

Bases: `fobi.base.BasePluginWidget`

Form handler plugin widget.

storage

alias of `FormHandlerPluginWidgetDataStorage`

`fobi.base.assemble_form_field_widget_class(base_class, plugin)`

Assemble form field widget class.

Finish this or remove.

#TODO

`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_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

- **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_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

- **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.

32.1.7 fobi.compat module

class `fobi.compat.User(*args, **kwargs)`

Bases: `django.contrib.auth.models.AbstractUser`

Users within the Django authentication system are represented by this model.

Username, password and email are required. Other fields are optional.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception User.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`User.formelement_set`

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.

`User.formentry_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.

`User.formhandler_set`

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.

User.formwizardentry_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.

User.get_next_by_date_joined (*moreargs, **morekwargs)

User.get_previous_by_date_joined (*moreargs, **morekwargs)

User.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.

User.id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

User.logentry_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.

User.registrationprofile

Accessor to the related object on the reverse side of a one-to-one relation.

In the example:

```
class Restaurant(Model):
    place = OneToOneField(Place, related_name='restaurant')
```

`place.restaurant` is a `ReverseOneToOneDescriptor` instance.

User.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.

User.user_permissions

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.

32.1.8 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.

32.1.9 fobi.constants module

32.1.10 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`.

32.1.11 fobi.data_structures module

class `fobi.data_structures.SortableDict` (*data=None*)

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 ()

iteritems ()

Iter items (internal method).

iterkeys ()

intervalues ()

Iter values (internal method).

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 (*k*, **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 ()

32.1.12 fobi.decorators module

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',
>>>     'fobi.delete_formentry',
>>>     'fobi.add_formelemententry',
>>>     'fobi.change_formelemententry',
>>>     'fobi.delete_formelemententry',
>>> ])
>>> def edit_dashboard(request):
>>>     # your code
```

`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
```

32.1.13 fobi.defaults module

32.1.14 fobi.discover module

`fobi.discover.autodiscover()`
 Auto-discovers files that should be found by fobi.

32.1.15 fobi.dynamic module

```
fobi.dynamic.assemble_form_class(form_entry,                                base_class=<class
                                'django.forms.forms.BaseForm'>, request=None, ori-
                                gin=None, origin_kwargs_update_func=None, ori-
                                gin_return_func=None, form_element_entries=None)
```

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).

32.1.16 fobi.exceptions module

exception `fobi.exceptions.BaseException`

Bases: `exceptions.Exception`

Base exception.

exception `fobi.exceptions.ImproperlyConfigured`

Bases: `fobi.exceptions.BaseException`

Improperly configured.

Exception raised when developer didn't configure/write the code properly.

exception `fobi.exceptions.InvalidRegistryItemType`

Bases: `exceptions.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.DoesNotExist`

Bases: `fobi.exceptions.BaseException`

Raised when something does not exist.

exception `fobi.exceptions.ThemeDoesNotExist`

Bases: `fobi.exceptions.DoesNotExist`

Raised when no theme with given uid can be found.

exception `fobi.exceptions.PluginDoesNotExist`

Bases: `fobi.exceptions.DoesNotExist`

Raised when no plugin with given uid can be found.

exception `fobi.exceptions.FormElementPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

Raised when no form element plugin with given uid can be found.

exception `fobi.exceptions.FormHandlerPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

Raised when no form handler plugin with given uid can be found.

exception `fobi.exceptions.NoDefaultThemeSet`

Bases: `fobi.exceptions.ImproperlyConfigured`

Raised when no active theme is chosen.

exception `fobi.exceptions.FormPluginError`

Bases: `fobi.exceptions.BaseException`

Base error for form elements and handlers.

exception `fobi.exceptions.FormElementPluginError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form element plugin error occurs.

exception `fobi.exceptions.FormHandlerPluginError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form handler plugin error occurs.

exception `fobi.exceptions.FormCallbackError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form callback error occurs.

32.1.17 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 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.

```

32.1.18 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.

```

32.1.19 fobi.forms module

```

class fobi.forms.FormEntryForm (*args, **kwargs)
    Bases: django.forms.models.ModelForm

    Form for fobi.models.FormEntry model.

    class Meta
        Meta class.

        fields = ('name', 'is_public', 'success_page_title', 'success_page_message', 'action')

        model
            alias of FormEntry

    FormEntryForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f156aa07b90>),
    FormEntryForm.clean_action()
        Validate the action (URL).

        Checks if URL exists.

```

```

    FormEntryForm.declared_fields = OrderedDict()

    FormEntryForm.media

class fobi.forms.FormFieldsetEntryForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm

    Form for fobi.models.FormFieldsetEntry model.

    class Meta
        Meta class.

        fields = ('name',)

        model
            alias of FormFieldsetEntry

    FormFieldsetEntryForm.base_fields = OrderedDict([('name', <django.forms.fields.CharField object at 0x7f156...
    FormFieldsetEntryForm.declared_fields = OrderedDict()

    FormFieldsetEntryForm.media

fobi.forms.FormElementEntryFormSet
    alias of FormElementEntryFormFormSet

class fobi.forms.BulkChangeFormElementPluginsForm(*args, **kwargs)
    Bases: fobi.forms.BaseBulkChangePluginsForm

    Bulk change form element plugins form.

    class Meta
        Meta class.

        fields = ['groups', 'groups_action', 'users', 'users_action']

        model
            alias of FormElement

    BulkChangeFormElementPluginsForm.base_fields = OrderedDict([('groups', <django.forms.models.ModelFormMu...
    BulkChangeFormElementPluginsForm.declared_fields = OrderedDict([('selected_plugins', <django.forms.fie...
    BulkChangeFormElementPluginsForm.media

class fobi.forms.BulkChangeFormHandlerPluginsForm(*args, **kwargs)
    Bases: fobi.forms.BaseBulkChangePluginsForm

    Bulk change form handler plugins form.

    class Meta
        Meta class.

        fields = ['groups', 'groups_action', 'users', 'users_action']

        model
            alias of FormHandler

    BulkChangeFormHandlerPluginsForm.base_fields = OrderedDict([('groups', <django.forms.models.ModelFormMu...
    BulkChangeFormHandlerPluginsForm.declared_fields = OrderedDict([('selected_plugins', <django.forms.fie...
    BulkChangeFormHandlerPluginsForm.media

```

```
class fobi.forms.ImportFormEntryForm(data=None, files=None, auto_id=u'id_%s', pre-
                                     fix=None, initial=None, error_class=<class
                                     'django.forms.utils.ErrorList'>, label_suffix=None,
                                     empty_permitted=False, field_order=None,
                                     use_required_attribute=None)
```

Bases: django.forms.forms.Form

Import form entry form.

```
base_fields = OrderedDict([('file', <django.forms.fields.FileField object at 0x7f156a799410>)])
```

```
declared_fields = OrderedDict([('file', <django.forms.fields.FileField object at 0x7f156a799410>)])
```

media

```
class fobi.forms.FormHandlerForm(data=None, files=None, auto_id=u'id_%s', prefix=None, ini-
                                  tial=None, error_class=<class 'django.forms.utils.ErrorList'>,
                                  label_suffix=None, empty_permitted=False, instance=None,
                                  use_required_attribute=None)
```

Bases: django.forms.models.ModelForm

FormHandler form.

```
class Meta
```

Meta class.

```
fields = ('users', 'groups')
```

```
model
```

alias of FormHandler

```
FormHandlerForm.base_fields = OrderedDict([('users', <django.forms.models.ModelMultipleChoiceField object at 0x7f156a799410>)])
```

```
FormHandlerForm.declared_fields = OrderedDict()
```

FormHandlerForm.media

```
class fobi.forms.FormHandlerEntryForm(data=None, files=None, auto_id=u'id_%s', pre-
                                       fix=None, initial=None, error_class=<class
                                       'django.forms.utils.ErrorList'>, label_suffix=None,
                                       empty_permitted=False, instance=None,
                                       use_required_attribute=None)
```

Bases: django.forms.models.ModelForm

FormHandlerEntry form.

```
class Meta
```

Meta class.

```
fields = ('form_entry', 'plugin_data', 'plugin_uid')
```

```
model
```

alias of FormHandlerEntry

```
FormHandlerEntryForm.base_fields = OrderedDict([('form_entry', <django.forms.models.ModelChoiceField object at 0x7f156a799410>)])
```

```
FormHandlerEntryForm.declared_fields = OrderedDict([('plugin_uid', <django.forms.fields.ChoiceField object at 0x7f156a799410>)])
```

FormHandlerEntryForm.media

32.1.20 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).

`fobi.helpers.do_slugify(s)`
 Slugify.

`fobi.helpers.lists_overlap(sub, main)`
 Check whether lists overlap.

`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.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.clean_dict(source, keys=[], values=[])`
 Removes given keys and values from dictionary.

Parameters

- **source** (*dict*) –
- **keys** (*iterable*) –
- **values** (*iterable*) –

Return dict

`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.ensure_unique_filename(destination)`
 Makes sure filenames are never overwritten.

Parameters **destination** (*string*) –

Return 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.delete_file(image_file)`
 Delete file from disc.

`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.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.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.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.safe_text(text)`

Safe text (encode).

Return str

`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.update_plugin_data(entry, request=None)`

Update plugin data.

Update plugin data of a given entry.

`fobi.helpers.get_select_field_choices(raw_choices_data)`

Get select field choices.

Used in `radio`, `select` and other choice based fields.

Parameters `raw_choices_data` (*str*) –

Return list

`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`) –

`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`) –

`fobi.helpers.validate_submit_value_as` (*value*)

Validates the `SUBMIT_AS_VALUE`.

Parameters `value` (*str*) –

`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`

class `fobi.helpers.StrippedUser` (*user*)

Bases: `object`

Stripped user object.

email

Email.

get_full_name ()

Get full name.

get_short_name ()

Get short name.

get_username ()

Get username.

is_anonymous ()

Is anonymous.

class `fobi.helpers.StrippedRequest` (*request*)

Bases: `object`

Stripped request object.

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.

path

Path.

A string representing the full path to the requested page, not including the scheme or domain.

class fobi.helpers.JSONDataExporter(*data, filename*)

Bases: object

Exporting the data into JSON.

export()

Export.

export_to_json()

Export data to JSON.

32.1.21 fobi.models module

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

Meta class.

abstract = False

AbstractPluginModel.**get_registered_plugins()**

Get registered plugins.

AbstractPluginModel.**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.

`AbstractPluginModel.groups_list()`

Groups list.

Flat list (comma separated string) of groups allowed to use the plugin. Used in Django admin.

Return string

`AbstractPluginModel.plugin_uid_admin()`

Plugin uid admin.

Mainly used in admin.

`AbstractPluginModel.plugin_uid_code()`

Plugin uid code.

Mainly used in admin.

`AbstractPluginModel.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.

`AbstractPluginModel.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.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 FormElement.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`FormElement.get_registered_plugins()`

Add choices.

FormElement.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.

FormElement.id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

FormElement.objects = <django.db.models.manager.Manager object>

FormElement.plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

FormElement.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.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 FormHandler.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`FormHandler.get_registered_plugins()`

Add choices.

FormHandler.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.

`FormHandler.id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormHandler.objects = <django.db.models.manager.Manager object>`

`FormHandler.plugin_uid`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormHandler.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.AbstractPluginEntry(*args, **kwargs)`

Bases: `django.db.models.base.Model`

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`

Meta class.

abstract = False

`AbstractPluginEntry.form_entry`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`AbstractPluginEntry.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.

`AbstractPluginEntry.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`.

`AbstractPluginEntry.get_registered_plugins()`

Get registered plugins.

`AbstractPluginEntry.get_registry()`

Get registry.

`AbstractPluginEntry.plugin_data`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`AbstractPluginEntry.plugin_name()`

Plugin name.

`AbstractPluginEntry.plugin_uid_code()`

Plugin uid code.

Mainly used in admin.

class `fobi.models.FormWizardEntry(*args, **kwargs)`

Bases: `django.db.models.base.Model`

Form wizard entry.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `FormWizardEntry.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`FormWizardEntry.formentry_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.

`FormWizardEntry.get_absolute_url()`

Get absolute URL.

Absolute URL, which goes to the dashboard workspace page.

Return string

`FormWizardEntry.id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormWizardEntry.is_cloneable`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormWizardEntry.is_public`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormWizardEntry.name`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormWizardEntry.objects = <django.db.models.manager.Manager object>`

`FormWizardEntry.slug`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormWizardEntry.user`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`FormWizardEntry.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.FormEntry(*args, **kwargs)`

Bases: `django.db.models.base.Model`

Form entry.

Properties

- `user` (`django.contrib.auth.models.User`): User owning the plugin.
- `wizard` (`str`): Form wizard to which the form entry belongs to.
- `name` (`str`): Form name.
- `slug` (`str`): Form slug.
- `description` (`str`): Form description.
- `is_public` (`bool`): If set to True, is visible to public.
- `is_clonable` (`bool`): If set to True, is clonable.
- `position` (`int`): Ordering position in the wizard.

`exception DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

`exception FormEntry.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`FormEntry.action`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

FormEntry.created

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

FormEntry.form_wizard_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

FormEntry.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.

FormEntry.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.

FormEntry.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.

FormEntry.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.

FormEntry.get_absolute_url()

Get absolute URL.

Absolute URL, which goes to the dashboard workspace page.

Return string

`FormEntry.id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormEntry.is_cloneable`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormEntry.is_public`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormEntry.name`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormEntry.objects = <django.db.models.manager.Manager object>`

`FormEntry.position`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormEntry.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.

`FormEntry.slug`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormEntry.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.

`FormEntry.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.

`FormEntry.updated`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormEntry.user`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`FormEntry.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.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 FormElementEntry.MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

`FormElementEntry.form_entry`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`FormElementEntry.form_fieldset_entry`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`FormElementEntry.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.

`FormElementEntry.get_registered_plugins()`

Gets registered plugins.

`FormElementEntry.get_registry()`

Get registry.

`FormElementEntry.id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormElementEntry.objects` = `<django.db.models.manager.Manager object>`

`FormElementEntry.plugin_uid`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormElementEntry.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.FormFieldsetEntry(*args, **kwargs)`

Bases: `django.db.models.base.Model`

Form fieldset entry.

exception `DoesNotExist`

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception `FormFieldsetEntry.MultipleObjectsReturned`

Bases: `django.core.exceptions.MultipleObjectsReturned`

`FormFieldsetEntry.form_entry`

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

`child.parent` is a `ForwardManyToOneDescriptor` instance.

`FormFieldsetEntry.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.

`FormFieldsetEntry.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.id`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormFieldsetEntry.is_repeatable`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormFieldsetEntry.name`

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

`FormFieldsetEntry.objects` = `<django.db.models.manager.Manager object>`

class `fobi.models.FormHandlerEntry(*args, **kwargs)`

Bases: `fobi.models.AbstractPluginEntry`

Form handler 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.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception FormHandlerEntry.MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

FormHandlerEntry.form_entry

Accessor to the related object on the forward side of a many-to-one or one-to-one relation.

In the example:

```
class Child(Model):
    parent = ForeignKey(Parent, related_name='children')
```

child.parent is a ForwardManyToOneDescriptor instance.

FormHandlerEntry.get_registered_plugins()

Gets registered plugins.

FormHandlerEntry.get_registry()

Get registry.

FormHandlerEntry.id

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

FormHandlerEntry.objects = <django.db.models.manager.Manager object>

FormHandlerEntry.plugin_uid

A wrapper for a deferred-loading field. When the value is read from this object the first time, the query is executed.

32.1.22 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.
- *DEBUG*

32.1.23 fobi.test module

32.1.24 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.get_allowed_plugin_uids(PluginModel, user)`

Get allowed plugins uids for user given.

Parameters

- **PluginModel** (`fobi.models.AbstractPluginModel`) – Subclass of `fobi.models.AbstractPluginModel`.
- **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, autodiscovers 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_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.sync_plugins()`

Sync registered plugins.

Syncs the registered plugin list with data in `fobi.models.FormFieldPluginModel` and `fobi.models.FormHandlerPluginModel`.

`fobi.utils.get_allowed_form_element_plugin_uids(user)`

Get allowed form element plugin uids.

`fobi.utils.get_user_form_element_plugins(user)`

Get user form element plugins.

`fobi.utils.get_allowed_form_handler_plugin_uids(user)`

Get allowed 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_plugin_uids(user)`

Get user form handler plugin uids.

`fobi.utils.get_user_plugins_grouped(get_allowed_plugin_uids_func, get_registered_plugins_grouped_func, registry, user, sort_items=True)`

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`) –

Return dict

`fobi.utils.get_user_form_element_plugins_grouped(user)`

Get user form element plugins grouped.

`fobi.utils.get_user_form_handler_plugins_grouped(user)`

Get user form handler plugins grouped.

32.1.25 fobi.validators module

`fobi.validators.url_exists(url, local=False)`

Check if URL exists.

Parameters

- **url** (`str`) –
- **local** (`bool`) –

Return bool

32.1.26 fobi.views module

Views.

`fobi.views.create_form_entry(request, *args, **kwargs)`

Create form entry.

Parameters

- **request** (`django.http.HttpRequest`) –
- **theme** (`fobi.base.BaseTheme`) – Theme instance.
- **template_name** (`str`) –

Return `django.http.HttpResponse`

`fobi.views.edit_form_entry(request, *args, **kwargs)`

Edit form entry.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry_id** (`int`) –
- **theme** (`fobi.base.BaseTheme`) – Theme instance.
- **template_name** (`str`) –

Return `django.http.HttpResponse`

`fobi.views.delete_form_entry(request, *args, **kwargs)`

Delete form entry.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry_id** (`int`) –
- **template_name** (`string`) –

Return `django.http.HttpResponse`

`fobi.views.add_form_element_entry(request, *args, **kwargs)`

Add form element entry.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry_id** (`int`) –
- **form_element_plugin_uid** (`int`) –
- **theme** (`fobi.base.BaseTheme`) – Theme instance.
- **template_name** (`string`) –

Return `django.http.HttpResponse`

`fobi.views.edit_form_element_entry(request, *args, **kwargs)`

Edit form element entry.

Parameters

- **request** (`django.http.HttpRequest`) –

- **form_element_entry_id**(*int*) –
- **fobi.base.BaseTheme** – Theme instance.
- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.delete_form_element_entry(request, *args, **kwargs)`
Delete form element entry.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_element_entry_id**(*int*) –

Return `django.http.HttpResponse`

`fobi.views.add_form_handler_entry(request, *args, **kwargs)`
Add form handler entry.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_entry_id**(*int*) –
- **form_handler_plugin_uid**(*int*) –
- **theme** (*fobi.base.BaseTheme*) – Theme instance.
- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.edit_form_handler_entry(request, *args, **kwargs)`
Edit form handler entry.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_handler_entry_id**(*int*) –
- **theme** (*fobi.base.BaseTheme*) – Theme instance.
- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.delete_form_handler_entry(request, *args, **kwargs)`
Delete form handler entry.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_handler_entry_id**(*int*) –

Return `django.http.HttpResponse`

`fobi.views.dashboard(request, *args, **kwargs)`
Dashboard.

Parameters

- **request** (*django.http.HttpRequest*) –
- **theme** (*fobi.base.BaseTheme*) – Theme instance.

- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.view_form_entry(request, form_entry_slug, theme=None, template_name=None)`
View created form.

Parameters

- **request**(`django.http.HttpRequest`) –
- **form_entry_slug**(*string*) –
- **theme**(`fobi.base.BaseTheme`) – Theme instance.
- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.form_entry_submitted(request, form_entry_slug=None, template_name=None)`
Form entry submitted.

Parameters

- **request**(`django.http.HttpRequest`) –
- **form_entry_slug**(*string*) –
- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.export_form_entry(request, *args, **kwargs)`
Export form entry to JSON.

Parameters

- **request**(`django.http.HttpRequest`) –
- **form_entry_id**(*int*) –
- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.import_form_entry(request, *args, **kwargs)`
Import form entry.

Parameters

- **request**(`django.http.HttpRequest`) –
- **template_name**(*string*) –

Return `django.http.HttpResponse`

`fobi.views.form_importer(request, *args, **kwargs)`
Form importer.

Parameters

- **request**(`django.http.HttpRequest`) –
- **form_importer_plugin_uid**(*str*) –
- **template_name**(*str*) –

32.1.27 fobi.widgets module

```
class fobi.widgets.NumberInput (attrs=None)
    Bases: django.forms.widgets.TextInput

    input_type = u'number'

    media

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)
```

media

32.1.28 Module contents

32.2 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)

32.2.1 Part 1: standard Django installation

Example project code available [here](#).

Installation and configuration

Install the package in your environment.

```
pip install django-fobi
```

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_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',
```

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',
```

```

# ...
# 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_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',

# ...
)

```

TEMPLATE_CONTEXT_PROCESSORS

Add `django.core.context_processors.request` and `fobi.context_processors.theme` to `TEMPLATE_CONTEXT_PROCESSORS` of your *settings* module.

```

TEMPLATE_CONTEXT_PROCESSORS = (
    # ...

```

```
"django.core.context_processors.request",
"fobi.context_processors.theme", # Obligatory
"fobi.context_processors.dynamic_values", # Optional
# ...
)
```

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')),

    # ...
]
```

Update the database

1. First you should be syncing/migrating the database. Depending on your Django version and migration app, this step may vary. Typically as follows:

```
$ ./manage.py syncdb
$ ./manage.py migrate --fake-initial
```

2. Sync installed fobi plugins. Go to terminal and type the following command.

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

Specify the active theme

Specify the default theme in your *settings* module.

```
FOBI_DEFAULT_THEME = 'bootstrap3'
```

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`

32.2.2 Part 2: Integration with DjangoCMS

Coming soon...

Indices and tables

- `genindex`
- `modindex`
- `search`

- [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

f

fobi, 227
 fobi.admin, 181
 fobi.app, 184
 fobi.apps, 184
 fobi.base, 184
 fobi.compat, 197
 fobi.conf, 199
 fobi.constants, 199
 fobi.context_processors, 199
 fobi.contrib, 174
 fobi.contrib.apps, 104
 fobi.contrib.apps.djangocms_integration, 100
 fobi.contrib.apps.djangocms_integration.apps, 99
 fobi.contrib.apps.djangocms_integration.conf, 100
 fobi.contrib.apps.djangocms_integration.defaults, 100
 fobi.contrib.apps.djangocms_integration.helpers, 100
 fobi.contrib.apps.djangocms_integration.settings, 100
 fobi.contrib.apps.feincms_integration, 103
 fobi.contrib.apps.feincms_integration.apps, 100
 fobi.contrib.apps.feincms_integration.conf, 101
 fobi.contrib.apps.feincms_integration.defaults, 101
 fobi.contrib.apps.feincms_integration.helpers, 101
 fobi.contrib.apps.feincms_integration.settings, 101
 fobi.contrib.apps.feincms_integration.widgets, 101
 fobi.contrib.apps.mezzanine_integration, 104
 fobi.contrib.apps.mezzanine_integration.apps, 103
 fobi.contrib.apps.mezzanine_integration.conf, 103
 fobi.contrib.apps.mezzanine_integration.defaults, 103
 fobi.contrib.apps.mezzanine_integration.helpers, 103
 fobi.contrib.apps.mezzanine_integration.settings, 104
 fobi.contrib.plugins, 161
 fobi.contrib.plugins.form_elements, 151
 fobi.contrib.plugins.form_elements.content, 110
 fobi.contrib.plugins.form_elements.content.content, 107
 fobi.contrib.plugins.form_elements.content.content.content, 105
 fobi.contrib.plugins.form_elements.content.content.content.content, 105
 fobi.contrib.plugins.form_elements.content.content.content.content.content, 105
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content, 106
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content, 106
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content.content, 107
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content.content.content, 108
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content.content.content.content, 107
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content.content.content.content, 107
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content.content.content.content.content, 108
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content.content.content.content.content, 108
 fobi.contrib.plugins.form_elements.content.content.content.content.content.content.content.content.content.content.content.content, 110

```

fobi.contrib.plugins.form_elements.conf.fobicontribvideogappsform_elements.fields.datetime,
108 116
fobi.contrib.plugins.form_elements.conf.fobicontribvideogcnafform_elements.fields.datetime
109 115
fobi.contrib.plugins.form_elements.conf.fobicontribvideogdesafform_elements.fields.datetime
109 115
fobi.contrib.plugins.form_elements.conf.fobicontribvideogfobiformelements.fields.datetime
109 116
fobi.contrib.plugins.form_elements.conf.fobicontribvideogformsform_elements.fields.datetime
109 116
fobi.contrib.plugins.form_elements.conf.fobicontribvideogsetforms_elements.fields.decimal,
110 118
fobi.contrib.plugins.form_elements.fieldsfobi.contrib.plugins.form_elements.fields.decimal.
145 117
fobi.contrib.plugins.form_elements.fieldsfobicontrib.plugins.form_elements.fields.decimal.
111 117
fobi.contrib.plugins.form_elements.fieldsfobicontribappsplugins.form_elements.fields.decimal.
110 117
fobi.contrib.plugins.form_elements.fieldsfobicontribfobipluginsformelements.fields.email,
110 119
fobi.contrib.plugins.form_elements.fieldsfobicontribformsplugins.form_elements.fields.email.app
111 118
fobi.contrib.plugins.form_elements.fieldsfobicontribcheckboxpluginsformelements.fields.email.fob
112 118
fobi.contrib.plugins.form_elements.fieldsfobicontribcheckboxpluginsformelements.fields.email.fob
111 118
fobi.contrib.plugins.form_elements.fieldsfobicontribcheckboxpluginsformelements.fields.file,
111 120
fobi.contrib.plugins.form_elements.fieldsfobicontribcheckboxpluginsformelements.fields.file.app
111 119
fobi.contrib.plugins.form_elements.fieldsfobicontribcheckboxpluginsformelements.fields.file.defa
111 119
fobi.contrib.plugins.form_elements.fieldsfobicontribcheckboxpluginsformelements.fields.file.defa
112 119
fobi.contrib.plugins.form_elements.fieldsfobicontribcheckboxpluginsformelements.fields.file.fob
112 119
fobi.contrib.plugins.form_elements.fieldsfobicontrib.plugins.form_elements.fields.file.form
114 120
fobi.contrib.plugins.form_elements.fieldsfobicontribappsplugins.form_elements.fields.file.sett
113 120
fobi.contrib.plugins.form_elements.fieldsfobicontribfobipluginsform_elements.fields.float,
113 121
fobi.contrib.plugins.form_elements.fieldsfobicontribformsplugins.form_elements.fields.float.app
113 121
fobi.contrib.plugins.form_elements.fieldsfobicontribwidgetsplugins.form_elements.fields.float.fob
114 121
fobi.contrib.plugins.form_elements.fieldsfobicontribdropdwnplugins.form_elements.fields.float.fob
115 121
fobi.contrib.plugins.form_elements.fieldsfobicontribdropdwnappsform_elements.fields.hidden,
114 123
fobi.contrib.plugins.form_elements.fieldsfobicontribdropdwnfobiformelements.fields.hidden.ap
114 122
fobi.contrib.plugins.form_elements.fieldsfobicontribdropdwnformsform_elements.fields.hidden.f
115 122

```

fobi.contrib.plugins.form_elements.fields.hidden	122	fobi.contrib.plugins.form_elements.fields.radio	129
fobi.contrib.plugins.form_elements.fields.input	124	fobi.contrib.plugins.form_elements.fields.select	130
fobi.contrib.plugins.form_elements.fields.password	123	fobi.contrib.plugins.form_elements.fields.regex	131
fobi.contrib.plugins.form_elements.fields.checkbox	123	fobi.contrib.plugins.form_elements.fields.regex.app	130
fobi.contrib.plugins.form_elements.fields.datetime	124	fobi.contrib.plugins.form_elements.fields.regex.fob	130
fobi.contrib.plugins.form_elements.fields.float	124	fobi.contrib.plugins.form_elements.fields.regex.for	130
fobi.contrib.plugins.form_elements.fields.integer	125	fobi.contrib.plugins.form_elements.fields.select	133
fobi.contrib.plugins.form_elements.fields.select	124	fobi.contrib.plugins.form_elements.fields.select.app	131
fobi.contrib.plugins.form_elements.fields.select.date	125	fobi.contrib.plugins.form_elements.fields.select.co	131
fobi.contrib.plugins.form_elements.fields.select.datetime	125	fobi.contrib.plugins.form_elements.fields.select.d	132
fobi.contrib.plugins.form_elements.fields.select.file	126	fobi.contrib.plugins.form_elements.fields.select.f	132
fobi.contrib.plugins.form_elements.fields.select.file.app	126	fobi.contrib.plugins.form_elements.fields.select.f	132
fobi.contrib.plugins.form_elements.fields.select.file.s	126	fobi.contrib.plugins.form_elements.fields.select_m	133
fobi.contrib.plugins.form_elements.fields.select_m	126	fobi.contrib.plugins.form_elements.fields.select_m	134
fobi.contrib.plugins.form_elements.fields.select_m	127	fobi.contrib.plugins.form_elements.fields.select_m	133
fobi.contrib.plugins.form_elements.fields.select_m	126	fobi.contrib.plugins.form_elements.fields.select_m	133
fobi.contrib.plugins.form_elements.fields.select_m	127	fobi.contrib.plugins.form_elements.fields.select_m	133
fobi.contrib.plugins.form_elements.fields.select_m	127	fobi.contrib.plugins.form_elements.fields.select_m	135
fobi.contrib.plugins.form_elements.fields.select_m	127	fobi.contrib.plugins.form_elements.fields.select_m	134
fobi.contrib.plugins.form_elements.fields.select_m	130	fobi.contrib.plugins.form_elements.fields.select_m	135
fobi.contrib.plugins.form_elements.fields.select_m	128	fobi.contrib.plugins.form_elements.fields.select_m	135
fobi.contrib.plugins.form_elements.fields.select_m	128	fobi.contrib.plugins.form_elements.fields.select_m	135
fobi.contrib.plugins.form_elements.fields.select_m	129	fobi.contrib.plugins.form_elements.fields.select_m	135
fobi.contrib.plugins.form_elements.fields.select_m	129	fobi.contrib.plugins.form_elements.fields.select_m	137

```

fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginappform_elements.fields.textarea,
135 143
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginonform_elements.fields.textarea
136 142
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.plugindefaultform_elements.fields.textarea
136 142
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginfobiform_elements.fields.textarea
136 142
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginformform_elements.fields.time,
136 144
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.time.app
137 143
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.time.fobi
139 143
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.time.form
137 144
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url,
137 145
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.app,
138 144
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.fobi
138 145
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.form
138 145
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity,
138 150
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.captcha
140 147
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.app
139 146
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.conf
139 146
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.default
139 146
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
139 148
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
139 147
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
141 147
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
140 147
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
140 147
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
140 148
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
142 148
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
141 148
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
141 150
fobi.contrib.plugins.form_elements.fields.fobiselectmultib.pluginsetform_elements.fields.url.setsengsity.honeyp
141 149

```

Python Module Index	241
----------------------------	------------

`fobi.views`, [224](#)
`fobi.widgets`, [227](#)

A

A

- abstract (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget.Meta attribute), 101
- abstract (fobi.models.AbstractPluginEntry.Meta attribute), 214
- abstract (fobi.models.AbstractPluginModel.Meta attribute), 211
- AbstractPluginEntry (class in fobi.models), 214
- AbstractPluginEntry.Meta (class in fobi.models), 214
- AbstractPluginModel (class in fobi.models), 211
- AbstractPluginModel.Meta (class in fobi.models), 211
- action (fobi.models.FormEntry attribute), 216
- actions (fobi.admin.FormElementAdmin attribute), 181
- actions (fobi.admin.FormHandlerAdmin attribute), 183
- actions (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDataEntryAdmin attribute), 151
- add_form_element_entry() (in module fobi.views), 224
- add_form_element_entry_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 163
- add_form_element_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes attribute), 166
- add_form_element_entry_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 170
- add_form_element_entry_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173
- add_form_element_entry_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 163
- add_form_element_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes attribute), 166
- add_form_element_entry_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 170
- add_form_element_entry_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173
- add_form_template (fobi.base.BasePlugin attribute), 186
- admin_change_url() (in module fobi.helpers), 209
- all_permissions_required() (in module fobi.decorators), 202
- allow_multiple (fobi.base.FormHandlerPlugin attribute), 190
- allow_multiple (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers attribute), 152
- any_permission_required() (in module fobi.decorators), 202
- app_config() (in module fobi.app), 184
- app_label (fobi.admin.BasePluginModelAdmin.Meta attribute), 181
- app_label (fobi.admin.FormElementEntryAdmin.Meta attribute), 182

[app_label \(fobi.admin.FormEntryAdmin.Meta attribute\), 182](#)
[app_label \(fobi.admin.FormFieldsetEntryAdmin.Meta attribute\), 183](#)
[app_label \(fobi.admin.FormHandlerEntryAdmin.Meta attribute\), 183](#)
[app_label \(fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget.Meta attribute\), 101](#)
[app_label \(fobi.contrib.plugins.form_handlers.db_store.admin_saved_form_data_entry_plugins.Meta attribute\), 151](#)
[app_name\(\) \(in module fobi.app\), 184](#)
[as_text\(\) \(fobi.form_utils.ErrorDict method\), 205](#)
[as_text\(\) \(fobi.form_utils.ErrorList method\), 205](#)
[assemble_form_class\(\) \(in module fobi.dynamic\), 203](#)
[assemble_form_field_widget_class\(\) \(in module fobi.base\), 196](#)
[autodiscover\(\) \(in module fobi.discover\), 202](#)

B

[base_bulk_change_plugins\(\) \(in module fobi.admin\), 184](#)
[base_edit_template \(fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute\), 166](#)
[base_edit_template \(fobi.contrib.themes.simple.fobi_themes.simple_theme attribute\), 173](#)
[base_fields \(fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm attribute\), 106](#)
[base_fields \(fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm attribute\), 108](#)
[base_fields \(fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm attribute\), 110](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleForm attribute\), 112](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm attribute\), 113](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateDropDownForm attribute\), 115](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm attribute\), 116](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm attribute\), 117](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm attribute\), 118](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm attribute\), 120](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm attribute\), 121](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm attribute\), 122](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.input.forms.InputForm attribute\), 124](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm attribute\), 125](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm attribute\), 128](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.radio.forms.RadioButtonForm attribute\), 129](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.regex.forms.RegexForm attribute\), 131](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.select.forms.SelectForm attribute\), 132](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectForm attribute\), 134](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMpttModelObjectForm attribute\), 135](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleForm attribute\), 137](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.forms.SelectMultipleModelObjectForm attribute\), 138](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_object.forms.SelectMultipleMpttModelObjectForm attribute\), 139](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm attribute\), 140](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm attribute\), 142](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm attribute\), 143](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm attribute\), 144](#)
[base_fields \(fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm attribute\), 145](#)
[base_fields \(fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaForm attribute\), 146](#)
[base_fields \(fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotForm attribute\), 148](#)
[base_fields \(fobi.contrib.plugins.form_elements.security.recaptcha.forms.ReCaptchaForm attribute\), 149](#)
[base_fields \(fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm attribute\), 156](#)
[base_fields \(fobi.contrib.plugins.form_handlers.mail.forms.MailForm attribute\), 158](#)
[base_fields \(fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpImporterForm attribute\), 160](#)
[base_fields \(fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpImporterForm attribute\), 160](#)
[base_fields \(fobi.forms.BulkChangeFormElementPluginsForm attribute\), 206](#)
[base_fields \(fobi.forms.BulkChangeFormHandlerPluginsForm attribute\), 206](#)
[base_fields \(fobi.forms.FormEntryForm attribute\), 205](#)
[base_fields \(fobi.forms.FormFieldsetEntryForm attribute\), 206](#)
[base_fields \(fobi.forms.FormHandlerEntryForm attribute\), 207](#)
[base_fields \(fobi.forms.FormHandlerForm attribute\), 207](#)
[base_fields \(fobi.forms.ImportFormEntryForm attribute\), 207](#)
[base_template \(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute\), 163](#)

base_template (fobi.contrib.themes.djangocms_admin_style.Theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 166

base_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 170

base_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173

base_view_template (fobi.contrib.themes.djangocms_admin_style.Theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 166

base_view_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173

BaseDataStorage (class in fobi.base), 184

BaseDatePluginWidget (class in fobi.contrib.plugins.form_elements.fields.date.widgets), 114

BaseDateTimePluginWidget (class in fobi.contrib.plugins.form_elements.fields.datetime.widgets), 116

BaseDbStorePluginWidget (class in fobi.contrib.plugins.form_handlers.db_store.widgets), 154

BaseDummyPluginWidget (class in fobi.contrib.plugins.form_elements.test.dummy.widgets), 150

BaseException, 203

BaseFormFieldPluginForm (class in fobi.base), 195

BaseFormImporter (class in fobi.form_importers), 204

BasePlugin (class in fobi.base), 185

BasePluginForm (class in fobi.base), 185

BasePluginModelAdmin (class in fobi.admin), 181

BasePluginModelAdmin.Meta (class in fobi.admin), 181

BaseRegistry (class in fobi.base), 191

BooleanRadioSelect (class in fobi.widgets), 227

BooleanSelectPlugin (class in fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements), 110

Bootstrap3Theme (class in fobi.contrib.themes.bootstrap3.fobi_themes), 163

bulk_change_form_element_plugins() (in module fobi.admin), 184

bulk_change_form_handler_plugins() (in module fobi.admin), 184

bulk_change_plugins() (fobi.admin.BasePluginModelAdmin method), 181

BulkChangeFormElementPluginsForm (class in fobi.forms), 206

BulkChangeFormElementPluginsForm.Meta (class in fobi.forms), 206

BulkChangeFormHandlerPluginsForm (class in fobi.forms), 206

BulkChangeFormHandlerPluginsForm.Meta (class in fobi.forms), 206

callback() (fobi.base.FormCallback method), 191

can_redirect (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 101

can_redirect (fobi.integration.processors.IntegrationProcessor attribute), 175

CaptchaInputForm (class in fobi.contrib.plugins.form_elements.security.captcha.forms), 146

CaptchaInputPlugin (class in fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements), 146

CheckboxSelectMultipleInputForm (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements), 112

CheckboxSelectMultipleInputPlugin (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements), 111

ClassProperty (class in fobi.base), 192

classproperty (in module fobi.base), 193

clean() (fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotForm method), 147

clean_action() (fobi.forms.FormEntryForm method), 205

clean_dict() (in module fobi.helpers), 208

clean_initial() (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements method), 112

clean_initial() (fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm method), 114

clean_initial() (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm method), 116

clean_initial() (fobi.contrib.plugins.form_elements.fields.radio.forms.RadioForm method), 129

clean_initial() (fobi.contrib.plugins.form_elements.fields.select.forms.SelectForm method), 132

clean_initial() (fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleForm method), 137

clean_initial() (fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm method), 144

cleans_up_after_itself (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserTest attribute), 179

clear() (fobi.data_structures.SortableDict method), 200

clone_file() (in module fobi.contrib.plugins.form_elements.content.content_image.helper), 106

clone_file() (in module fobi.helpers), 208

clone_plugin_data() (fobi.base.BasePlugin method), 186

clone_plugin_data() (fobi.contrib.plugins.form_elements.content.content_image.helper method), 105

code (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField attribute), 157

collect_plugin_media() (in module fobi.base), 194

combine_dicts() (in module fobi.helpers), 209

Command (class in fobi.management.commands.fobi_find_broken_entries), 175

Command (class in fobi.management.commands.fobi_migrate), 175	Config (class in fobi.contrib.plugins.form_elements.fields.select.apps), 131
Command (class in fobi.management.commands.fobi_sync_plugins), 176	Config (class in fobi.contrib.plugins.form_elements.fields.select_model_obj), 133
Command (class in fobi.management.commands.fobi_update_config), 176	Config (class in fobi.contrib.plugins.form_elements.fields.select_mptt_model_obj), 134
Config (class in fobi.apps), 184	Config (class in fobi.contrib.plugins.form_elements.fields.select_multiple.apps), 135
Config (class in fobi.contrib.apps.djangocms_integration.apps), 99	Config (class in fobi.contrib.plugins.form_elements.fields.select_multiple_model_obj), 137
Config (class in fobi.contrib.apps.feincms_integration.apps), 100	Config (class in fobi.contrib.plugins.form_elements.fields.select_multiple_model_obj), 139
Config (class in fobi.contrib.apps.mezzanine_integration.apps), 103	Config (class in fobi.contrib.plugins.form_elements.fields.slug.apps), 140
Config (class in fobi.contrib.plugins.form_elements.content.content_image.apps), 105	Config (class in fobi.contrib.plugins.form_elements.fields.text.apps), 141
Config (class in fobi.contrib.plugins.form_elements.content.content_text.apps), 107	Config (class in fobi.contrib.plugins.form_elements.fields.textarea.apps), 142
Config (class in fobi.contrib.plugins.form_elements.content.content_video.apps), 108	Config (class in fobi.contrib.plugins.form_elements.fields.time.apps), 143
Config (class in fobi.contrib.plugins.form_elements.fields.boolean.apps), 110	Config (class in fobi.contrib.plugins.form_elements.fields.url.apps), 144
Config (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps), 111	Config (class in fobi.contrib.plugins.form_elements.security.captcha.apps), 146
Config (class in fobi.contrib.plugins.form_elements.fields.date.apps), 113	Config (class in fobi.contrib.plugins.form_elements.security.honeypot.apps), 147
Config (class in fobi.contrib.plugins.form_elements.fields.date_drop_down.apps), 114	Config (class in fobi.contrib.plugins.form_elements.security.recaptcha.apps), 149
Config (class in fobi.contrib.plugins.form_elements.fields.datetime.apps), 115	Config (class in fobi.contrib.plugins.form_elements.test.dummy.apps), 150
Config (class in fobi.contrib.plugins.form_elements.fields.decimal.apps), 117	Config (class in fobi.contrib.plugins.form_handlers.db_store.apps), 151
Config (class in fobi.contrib.plugins.form_elements.fields.email.apps), 118	Config (class in fobi.contrib.plugins.form_handlers.http_repost.apps), 155
Config (class in fobi.contrib.plugins.form_elements.fields.file.apps), 119	Config (class in fobi.contrib.plugins.form_handlers.mail.apps), 156
Config (class in fobi.contrib.plugins.form_elements.fields.float.apps), 121	Config (class in fobi.contrib.plugins.form_importers.mailchimp_importer.apps), 159
Config (class in fobi.contrib.plugins.form_elements.fields.hidden.apps), 122	Config (class in fobi.contrib.themes.bootstrap3.apps), 163
Config (class in fobi.contrib.plugins.form_elements.fields.infinite_scroll.apps), 123	Config (class in fobi.contrib.themes.bootstrap3.widgets.form_elements.date_picker.apps), 161
Config (class in fobi.contrib.plugins.form_elements.fields.infinite_scroll.apps), 124	Config (class in fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_picker.apps), 162
Config (class in fobi.contrib.plugins.form_elements.fields.ip_address.apps), 126	Config (class in fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy.apps), 163
Config (class in fobi.contrib.plugins.form_elements.fields.null_boolean.apps), 126	Config (class in fobi.contrib.themes.djangocms_admin_style_theme.apps), 166
Config (class in fobi.contrib.plugins.form_elements.fields.password.apps), 127	Config (class in fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.password.apps), 165
Config (class in fobi.contrib.plugins.form_elements.fields.radio.apps), 128	Config (class in fobi.contrib.themes.foundation5.apps), 170
Config (class in fobi.contrib.plugins.form_elements.fields.regexp.apps), 130	Config (class in fobi.contrib.themes.foundation5.widgets.form_elements.date_picker.apps), 168

[delete_form_element_entry\(\)](#) (in module `fobi.views`), [225](#)
[delete_form_entry\(\)](#) (in module `fobi.views`), [224](#)
[delete_form_handler_entry\(\)](#) (in module `fobi.views`), [225](#)
[delete_plugin_data\(\)](#) (`fobi.base.BasePlugin` method), [186](#)
[delete_plugin_data\(\)](#) (`fobi.contrib.plugins.form_elements.content_image_plugin.ContentImagePlugin` method), [105](#)
[description](#) (`fobi.base.BasePlugin` attribute), [186](#)
[description](#) (`fobi.form_importers.BaseFormImporter` attribute), [204](#)
[DjangoCMSAdminStyleTheme](#) (class in `fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes`), [166](#)
[do\(\)](#) (in module `fobi.tests.test_form_importers_mailchimp`), [180](#)
[do_slugify\(\)](#) (in module `fobi.helpers`), [207](#)
[DoesNotExist](#), [203](#)
[done\(\)](#) (`fobi.contrib.plugins.form_importers.mailchimp_importer.views.MailchimpImporterWizardView` method), [161](#)
[DummyPlugin](#) (class in `fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements`), [150](#)
[DummyPluginWidget](#) (class in `fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements`), [163](#)
[DummyPluginWidget](#) (class in `fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_elements`), [169](#)
[dynamic_values\(\)](#) (in module `fobi.context_processors`), [199](#)

E

[e](#) (`fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest` attribute), [179](#)
[edit_form_element_entry\(\)](#) (in module `fobi.views`), [224](#)
[edit_form_element_entry_ajax_template](#) (`fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme` attribute), [164](#)
[edit_form_element_entry_ajax_template](#) (`fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes` attribute), [166](#)
[edit_form_element_entry_ajax_template](#) (`fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme` attribute), [170](#)
[edit_form_element_entry_ajax_template](#) (`fobi.contrib.themes.simple.fobi_themes.SimpleTheme` attribute), [173](#)
[edit_form_element_entry_template](#) (`fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme` attribute), [164](#)
[edit_form_element_entry_template](#) (`fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes` attribute), [166](#)
[edit_form_element_entry_template](#) (`fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme` attribute), [166](#)

[edit_form_handler_entry_template](#) (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171
[edit_form_handler_entry_template](#) (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173
[edit_form_template](#) (fobi.base.BasePlugin attribute), 186
[email](#) (fobi.helpers.StrippedUser attribute), 210
[EmailInputForm](#) (class in fobi.contrib.plugins.form_elements.fields.email.forms), 118
[EmailInputPlugin](#) (class in fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements), 118
[embed_form_entry_submitted_ajax_template](#) (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164
[ensure_autodiscover\(\)](#) (in module fobi.base), 194
[ensure_autodiscover\(\)](#) (in module fobi.form_importers), 205
[ensure_unique_filename\(\)](#) (in module fobi.contrib.plugins.form_elements.content.content_image.helpers), 106
[ensure_unique_filename\(\)](#) (in module fobi.helpers), 208
[ErrorDict](#) (class in fobi.form_utils), 205
[ErrorList](#) (class in fobi.form_utils), 205
[export\(\)](#) (fobi.helpers.JSONDataExporter method), 211
[export_data\(\)](#) (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormEntryAdmin method), 151
[export_entries_icon_class](#) (fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget attribute), 154
[export_entries_icon_class](#) (fobi.contrib.themes.djangocms_admin_style_theme.widgets.fobi_form_elements.DbStorePluginWidget attribute), 165
[export_entries_icon_class](#) (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store.fobi_form_elements.DbStorePluginWidget attribute), 170
[export_entries_icon_class](#) (fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements.DbStorePluginWidget attribute), 172
[export_form_entry\(\)](#) (in module fobi.views), 226
[export_to_csv\(\)](#) (fobi.contrib.plugins.form_handlers.db_store.helpers.DataExporter method), 153
[export_to_json\(\)](#) (fobi.helpers.JSONDataExporter method), 211
[export_to_xls\(\)](#) (fobi.contrib.plugins.form_handlers.db_store.helpers.DataExporter method), 153
[extra](#) (fobi.admin.ModelFormEntryInlineAdmin attribute), 182
[extra](#) (fobi.admin.FormHandlerEntryInlineAdmin attribute), 183
[extract_field_properties\(\)](#) (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter method), 159
[extract_field_properties\(\)](#) (fobi.admin.ModelFormEntryAdmin attribute), 182
[fail_on_missing_plugin](#) (fobi.base.BaseRegistry attribute), 191
[fail_on_missing_plugin](#) (fobi.base.FormElementPluginRegistry attribute), 192
[fail_on_missing_plugin](#) (fobi.base.FormHandlerPluginRegistry attribute), 192
[fail_on_missing_plugin](#) (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter attribute), 160
[field_type_prop_name](#) (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter attribute), 160
[field_type_prop_name](#) (fobi.form_importers.BaseFormImporter attribute), 204
[fields](#) (fobi.admin.ModelFormEntryInlineAdmin attribute), 182
[fields](#) (fobi.admin.FormHandlerEntryInlineAdmin attribute), 183
[fields](#) (fobi.forms.BulkChangeFormElementPluginsForm.Meta attribute), 205
[fields](#) (fobi.forms.BulkChangeFormHandlerPluginsForm.Meta attribute), 206
[fields](#) (fobi.forms.FormFieldsetEntryForm.Meta attribute), 206
[fields](#) (fobi.forms.FormHandlerForm.Meta attribute), 207
[fields](#) (fobi.forms.FormHandlerForm.Meta attribute), 207
[fields_mapping](#) (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter attribute), 160
[fields_mapping](#) (fobi.form_importers.BaseFormImporter attribute), 204
[fieldsets](#) (fobi.admin.BasePluginModelAdmin attribute), 181
[fieldsets](#) (fobi.admin.ModelFormEntryAdmin attribute), 182
[fieldsets](#) (fobi.admin.FormEntryAdmin attribute), 182
[fieldsets](#) (fobi.admin.FormFieldsetEntryAdmin attribute), 183
[fieldsets](#) (fobi.admin.FormHandlerEntryAdmin attribute), 183
[fieldsets](#) (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormEntryAdmin attribute), 151
[FileInputForm](#) (class in fobi.contrib.plugins.form_elements.fields.file.forms), 110

FileInputPlugin (class in fobi.contrib.apps.mezzanine_integration.defaults (module), 119
 fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements), 119
 filter_horizontal (fobi.admin.BasePluginModelAdmin attribute), 181
 finalize() (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method), 101
 fire_form_callbacks() (in module fobi.base), 194
 firstof() (in module fobi.templatetags.future_compat), 177
 FloatInputForm (class in fobi.contrib.plugins.form_elements.content.content_image (module), 107
 fobi.contrib.plugins.form_elements.fields.float.forms), 121
 FloatInputPlugin (class in fobi.contrib.plugins.form_elements.content.content_image (module), 105
 fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements), 121
 fobi (module), 227
 fobi.admin (module), 181
 fobi.app (module), 184
 fobi.apps (module), 184
 fobi.base (module), 184
 fobi.compat (module), 197
 fobi.conf (module), 199
 fobi.constants (module), 199
 fobi.context_processors (module), 199
 fobi.contrib (module), 174
 fobi.contrib.apps (module), 104
 fobi.contrib.apps.djangocms_integration (module), 100
 fobi.contrib.apps.djangocms_integration.apps (module), 99
 fobi.contrib.apps.djangocms_integration.conf (module), 100
 fobi.contrib.apps.djangocms_integration.defaults (module), 100
 fobi.contrib.apps.djangocms_integration.helpers (module), 100
 fobi.contrib.apps.djangocms_integration.settings (module), 100
 fobi.contrib.apps.feincms_integration (module), 103
 fobi.contrib.apps.feincms_integration.apps (module), 100
 fobi.contrib.apps.feincms_integration.conf (module), 101
 fobi.contrib.apps.feincms_integration.defaults (module), 101
 fobi.contrib.apps.feincms_integration.helpers (module), 101
 fobi.contrib.apps.feincms_integration.settings (module), 101
 fobi.contrib.apps.feincms_integration.widgets (module), 101
 fobi.contrib.apps.mezzanine_integration (module), 104
 fobi.contrib.apps.mezzanine_integration.apps (module), 103
 fobi.contrib.apps.mezzanine_integration.conf (module), 103
 fobi.contrib.apps.mezzanine_integration.defaults (module), 103
 fobi.contrib.apps.mezzanine_integration.helpers (module), 103
 fobi.contrib.apps.mezzanine_integration.settings (module), 103
 fobi.contrib.apps.mezzanine_integration.widgets (module), 103
 fobi.contrib.plugins (module), 161
 fobi.contrib.plugins.form_elements (module), 151
 fobi.contrib.plugins.form_elements.content (module), 110
 fobi.contrib.plugins.form_elements.content.content_image (module), 107
 fobi.contrib.plugins.form_elements.content.content_image.apps (module), 105
 fobi.contrib.plugins.form_elements.content.content_image.conf (module), 105
 fobi.contrib.plugins.form_elements.content.content_image.defaults (module), 105
 fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements (module), 105
 fobi.contrib.plugins.form_elements.content.content_image.forms (module), 106
 fobi.contrib.plugins.form_elements.content.content_image.helpers (module), 106
 fobi.contrib.plugins.form_elements.content.content_image.settings (module), 107
 fobi.contrib.plugins.form_elements.content.content_text (module), 108
 fobi.contrib.plugins.form_elements.content.content_text.apps (module), 107
 fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements (module), 107
 fobi.contrib.plugins.form_elements.content.content_text.forms (module), 108
 fobi.contrib.plugins.form_elements.content.content_video (module), 110
 fobi.contrib.plugins.form_elements.content.content_video.apps (module), 108
 fobi.contrib.plugins.form_elements.content.content_video.conf (module), 109
 fobi.contrib.plugins.form_elements.content.content_video.defaults (module), 109
 fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements (module), 109
 fobi.contrib.plugins.form_elements.content.content_video.forms (module), 109
 fobi.contrib.plugins.form_elements.content.content_video.settings (module), 110
 fobi.contrib.plugins.form_elements.fields (module), 145
 fobi.contrib.plugins.form_elements.fields.boolean (module), 111
 fobi.contrib.plugins.form_elements.fields.boolean.apps (module), 110

fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements (module), 110	fobi.contrib.plugins.form_elements.fields.email (module), 119
fobi.contrib.plugins.form_elements.fields.boolean.forms (module), 111	fobi.contrib.plugins.form_elements.fields.email.apps (module), 118
fobi.contrib.plugins.form_elements.fields.checkbox_select_fobi_form_elements (module), 112	fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements (module), 118
fobi.contrib.plugins.form_elements.fields.checkbox_select_fobi_form_elements (module), 111	fobi.contrib.plugins.form_elements.fields.email.forms (module), 118
fobi.contrib.plugins.form_elements.fields.checkbox_select_fobi_form_elements (module), 111	fobi.contrib.plugins.form_elements.fields.file (module), 120
fobi.contrib.plugins.form_elements.fields.checkbox_select_fobi_form_elements (module), 111	fobi.contrib.plugins.form_elements.fields.file.apps (module), 119
fobi.contrib.plugins.form_elements.fields.checkbox_select_fobi_form_elements (module), 111	fobi.contrib.plugins.form_elements.fields.file.conf (module), 119
fobi.contrib.plugins.form_elements.fields.checkbox_select_fobi_form_elements (module), 112	fobi.contrib.plugins.form_elements.fields.file.defaults (module), 119
fobi.contrib.plugins.form_elements.fields.checkbox_select_fobi_form_elements (module), 112	fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements (module), 119
fobi.contrib.plugins.form_elements.fields.date (module), 114	fobi.contrib.plugins.form_elements.fields.file.forms (module), 120
fobi.contrib.plugins.form_elements.fields.date.apps (module), 113	fobi.contrib.plugins.form_elements.fields.file.settings (module), 120
fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements (module), 113	fobi.contrib.plugins.form_elements.fields.float (module), 121
fobi.contrib.plugins.form_elements.fields.date.forms (module), 113	fobi.contrib.plugins.form_elements.fields.float.apps (module), 121
fobi.contrib.plugins.form_elements.fields.date.widgets (module), 114	fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements (module), 121
fobi.contrib.plugins.form_elements.fields.date_drop_down (module), 115	fobi.contrib.plugins.form_elements.fields.float.forms (module), 121
fobi.contrib.plugins.form_elements.fields.date_drop_down.apps (module), 114	fobi.contrib.plugins.form_elements.fields.hidden (module), 123
fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements (module), 114	fobi.contrib.plugins.form_elements.fields.hidden.apps (module), 122
fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements (module), 115	fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements (module), 122
fobi.contrib.plugins.form_elements.fields.datetime (module), 116	fobi.contrib.plugins.form_elements.fields.hidden.forms (module), 122
fobi.contrib.plugins.form_elements.fields.datetime.apps (module), 115	fobi.contrib.plugins.form_elements.fields.input (module), 124
fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements (module), 115	fobi.contrib.plugins.form_elements.fields.input.apps (module), 123
fobi.contrib.plugins.form_elements.fields.datetime.forms (module), 116	fobi.contrib.plugins.form_elements.fields.input.constants (module), 123
fobi.contrib.plugins.form_elements.fields.datetime.widgets (module), 116	fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements (module), 124
fobi.contrib.plugins.form_elements.fields.decimal (module), 118	fobi.contrib.plugins.form_elements.fields.input.forms (module), 124
fobi.contrib.plugins.form_elements.fields.decimal.apps (module), 117	fobi.contrib.plugins.form_elements.fields.integer (module), 125
fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements (module), 117	fobi.contrib.plugins.form_elements.fields.integer.apps (module), 124
fobi.contrib.plugins.form_elements.fields.decimal.forms (module), 117	fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements (module), 125

fobi.contrib.plugins.form_elements.fields.integer.forms (module), 125	fobi.contrib.plugins.form_elements.fields.select.defaults (module), 132
fobi.contrib.plugins.form_elements.fields.ip_address (module), 126	fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements (module), 132
fobi.contrib.plugins.form_elements.fields.ip_address.apps (module), 126	fobi.contrib.plugins.form_elements.fields.select.forms (module), 132
fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements (module), 126	fobi.contrib.plugins.form_elements.fields.select.settings (module), 133
fobi.contrib.plugins.form_elements.fields.ip_address.forms (module), 126	fobi.contrib.plugins.form_elements.fields.select_model_object (module), 134
fobi.contrib.plugins.form_elements.fields.null_boolean (module), 127	fobi.contrib.plugins.form_elements.fields.select_model_object.apps (module), 133
fobi.contrib.plugins.form_elements.fields.null_boolean.apps (module), 126	fobi.contrib.plugins.form_elements.fields.select_model_object.conf (module), 133
fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements (module), 126	fobi.contrib.plugins.form_elements.fields.select_model_object.defaults (module), 133
fobi.contrib.plugins.form_elements.fields.null_boolean.forms (module), 127	fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements (module), 133
fobi.contrib.plugins.form_elements.fields.password (module), 128	fobi.contrib.plugins.form_elements.fields.select_model_object.forms (module), 134
fobi.contrib.plugins.form_elements.fields.password.apps (module), 127	fobi.contrib.plugins.form_elements.fields.select_model_object.settings (module), 134
fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements (module), 127	fobi.contrib.plugins.form_elements.fields.select_mptt_model_object (module), 135
fobi.contrib.plugins.form_elements.fields.password.forms (module), 127	fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps (module), 134
fobi.contrib.plugins.form_elements.fields.radio (module), 130	fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf (module), 135
fobi.contrib.plugins.form_elements.fields.radio.apps (module), 128	fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults (module), 135
fobi.contrib.plugins.form_elements.fields.radio.conf (module), 128	fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms (module), 135
fobi.contrib.plugins.form_elements.fields.radio.defaults (module), 129	fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.settings (module), 135
fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements (module), 129	fobi.contrib.plugins.form_elements.fields.select_multiple (module), 137
fobi.contrib.plugins.form_elements.fields.radio.forms (module), 129	fobi.contrib.plugins.form_elements.fields.select_multiple.apps (module), 135
fobi.contrib.plugins.form_elements.fields.radio.settings (module), 130	fobi.contrib.plugins.form_elements.fields.select_multiple.conf (module), 136
fobi.contrib.plugins.form_elements.fields.regex (module), 131	fobi.contrib.plugins.form_elements.fields.select_multiple.defaults (module), 136
fobi.contrib.plugins.form_elements.fields.regex.apps (module), 130	fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements (module), 136
fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements (module), 130	fobi.contrib.plugins.form_elements.fields.select_multiple.forms (module), 136
fobi.contrib.plugins.form_elements.fields.regex.forms (module), 130	fobi.contrib.plugins.form_elements.fields.select_multiple.settings (module), 137
fobi.contrib.plugins.form_elements.fields.select (module), 133	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects (module), 139
fobi.contrib.plugins.form_elements.fields.select.apps (module), 131	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps (module), 137
fobi.contrib.plugins.form_elements.fields.select.conf (module), 131	fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf (module), 137

fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice (module), 138	fobi.contrib.plugins.form_elements.fields.url.apps (module), 144
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 138	fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements (module), 145
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 138	fobi.contrib.plugins.form_elements.fields.url.forms (module), 145
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 138	fobi.contrib.plugins.form_elements.security (module), 150
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 140	fobi.contrib.plugins.form_elements.security.captcha (module), 147
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 139	fobi.contrib.plugins.form_elements.security.captcha.apps (module), 146
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 139	fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements (module), 146
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 139	fobi.contrib.plugins.form_elements.security.captcha.forms (module), 146
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 139	fobi.contrib.plugins.form_elements.security.honeypot (module), 148
fobi.contrib.plugins.form_elements.fields.select_multiple_multiple_choice.fobi_form_elements (module), 139	fobi.contrib.plugins.form_elements.security.honeypot.apps (module), 147
fobi.contrib.plugins.form_elements.fields.slug (module), 141	fobi.contrib.plugins.form_elements.security.honeypot.conf (module), 147
fobi.contrib.plugins.form_elements.fields.slug.apps (module), 140	fobi.contrib.plugins.form_elements.security.honeypot.defaults (module), 147
fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements (module), 140	fobi.contrib.plugins.form_elements.security.honeypot.fields (module), 147
fobi.contrib.plugins.form_elements.fields.slug.forms (module), 140	fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements (module), 148
fobi.contrib.plugins.form_elements.fields.text (module), 142	fobi.contrib.plugins.form_elements.security.honeypot.forms (module), 148
fobi.contrib.plugins.form_elements.fields.text.apps (module), 141	fobi.contrib.plugins.form_elements.security.honeypot.settings (module), 148
fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements (module), 141	fobi.contrib.plugins.form_elements.security.recaptcha (module), 150
fobi.contrib.plugins.form_elements.fields.text.forms (module), 141	fobi.contrib.plugins.form_elements.security.recaptcha.apps (module), 149
fobi.contrib.plugins.form_elements.fields.textarea (module), 143	fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements (module), 149
fobi.contrib.plugins.form_elements.fields.textarea.apps (module), 142	fobi.contrib.plugins.form_elements.security.recaptcha.forms (module), 149
fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements (module), 142	fobi.contrib.plugins.form_elements.test (module), 151
fobi.contrib.plugins.form_elements.fields.textarea.forms (module), 142	fobi.contrib.plugins.form_elements.test.dummy (module), 150
fobi.contrib.plugins.form_elements.fields.time (module), 144	fobi.contrib.plugins.form_elements.test.dummy.apps (module), 150
fobi.contrib.plugins.form_elements.fields.time.apps (module), 143	fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements (module), 150
fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements (module), 143	fobi.contrib.plugins.form_elements.test.dummy.widgets (module), 150
fobi.contrib.plugins.form_elements.fields.time.forms (module), 144	fobi.contrib.plugins.form_handlers (module), 159
fobi.contrib.plugins.form_elements.fields.url (module), 145	fobi.contrib.plugins.form_handlers.db_store (module), 155
	fobi.contrib.plugins.form_handlers.db_store.admin (module), 151

- fobi.contrib.plugins.form_handlers.db_store.apps (module), 151
- fobi.contrib.plugins.form_handlers.db_store.conf (module), 152
- fobi.contrib.plugins.form_handlers.db_store.defaults (module), 152
- fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers (module), 152
- fobi.contrib.plugins.form_handlers.db_store.helpers (module), 153
- fobi.contrib.plugins.form_handlers.db_store.models (module), 153
- fobi.contrib.plugins.form_handlers.db_store.settings (module), 154
- fobi.contrib.plugins.form_handlers.db_store.urls (module), 154
- fobi.contrib.plugins.form_handlers.db_store.views (module), 154
- fobi.contrib.plugins.form_handlers.db_store.widgets (module), 154
- fobi.contrib.plugins.form_handlers.http_repost (module), 156
- fobi.contrib.plugins.form_handlers.http_repost.apps (module), 155
- fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers (module), 155
- fobi.contrib.plugins.form_handlers.http_repost.forms (module), 155
- fobi.contrib.plugins.form_handlers.mail (module), 159
- fobi.contrib.plugins.form_handlers.mail.apps (module), 156
- fobi.contrib.plugins.form_handlers.mail.conf (module), 156
- fobi.contrib.plugins.form_handlers.mail.defaults (module), 157
- fobi.contrib.plugins.form_handlers.mail.fields (module), 157
- fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers (module), 157
- fobi.contrib.plugins.form_handlers.mail.forms (module), 158
- fobi.contrib.plugins.form_handlers.mail.helpers (module), 158
- fobi.contrib.plugins.form_handlers.mail.settings (module), 159
- fobi.contrib.plugins.form_handlers.mail.widgets (module), 159
- fobi.contrib.plugins.form_importers (module), 161
- fobi.contrib.plugins.form_importers.mailchimp_importer (module), 161
- fobi.contrib.plugins.form_importers.mailchimp_importer.apps (module), 159
- fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers (module), 159
- fobi.contrib.plugins.form_importers.mailchimp_importer.forms (module), 160
- fobi.contrib.plugins.form_importers.mailchimp_importer.views (module), 161
- fobi.contrib.themes (module), 174
- fobi.contrib.themes.bootstrap3 (module), 164
- fobi.contrib.themes.bootstrap3.apps (module), 163
- fobi.contrib.themes.bootstrap3.fobi_themes (module), 163
- fobi.contrib.themes.bootstrap3.widgets (module), 163
- fobi.contrib.themes.bootstrap3.widgets.form_elements (module), 163
- fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widgets (module), 162
- fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widgets (module), 161
- fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widgets (module), 162
- fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widgets (module), 162
- fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widgets (module), 162
- fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widgets (module), 162
- fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widgets (module), 162
- fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widgets (module), 163
- fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widgets (module), 163
- fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widgets (module), 163
- fobi.contrib.themes.djangocms_admin_style_theme (module), 167
- fobi.contrib.themes.djangocms_admin_style_theme.apps (module), 166
- fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes (module), 166
- fobi.contrib.themes.djangocms_admin_style_theme.widgets (module), 165
- fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers (module), 165
- fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers (module), 165
- fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers (module), 165
- fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers (module), 165
- fobi.contrib.themes.foundation5 (module), 171
- fobi.contrib.themes.foundation5.apps (module), 170
- fobi.contrib.themes.foundation5.fobi_themes (module), 170
- fobi.contrib.themes.foundation5.widgets (module), 170
- fobi.contrib.themes.foundation5.widgets.form_elements (module), 169

fobi.contrib.themes.foundation5.widgets.form_elements.data_foundation5_widgets_form_elements_data (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widgets_date (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widgets_datetime (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.fobi_update_plugin_data (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.file_upload_widget (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.folder_select_widget (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.integrate_plugins (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.template_tags_fobi_tags (module), 168

fobi.contrib.themes.foundation5.widgets.form_elements.duplicate_foundation5_widgets_duplicate (module), 169

fobi.contrib.themes.foundation5.widgets.form_elements.duration_foundation5_widgets_duration (module), 169

fobi.contrib.themes.foundation5.widgets.form_elements.dynamic_forms_test (module), 169

fobi.contrib.themes.foundation5.widgets.form_handlers fobi.tests.helpers (module), 170

fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget (module), 170

fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_functional_dynamic_forms_apps (module), 169

fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_functional_widgets_apps (module), 170

fobi.contrib.themes.simple (module), 174

fobi.contrib.themes.simple.apps (module), 172

fobi.contrib.themes.simple.fobi_themes (module), 173

fobi.contrib.themes.simple.widgets (module), 172

fobi.contrib.themes.simple.widgets.form_handlers (module), 172

fobi.contrib.themes.simple.widgets.form_handlers.db_store FobibrowserBuldDynamicFormsTest (class in module), 172

fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps FobicoreTest (class in fobi.tests.test_core), 179

fobi.contrib.themes.simple.widgets.form_handlers.db_store.FobiDataStructureTest (class in fobi.tests.test_sortable_dict), 180

fobi.data_structures (module), 200

fobi.decorators (module), 201

fobi.defaults (module), 202

fobi.discover (module), 202

fobi.dynamic (module), 203

fobi.exceptions (module), 203

fobi.form_importers (module), 204

fobi.form_utils (module), 205

fobi.forms (module), 205

fobi.helpers (module), 207

fobi.integration (module), 175

fobi.integration.helpers (module), 174

fobi.integration.processors (module), 174

fobi.management (module), 176

fobi.management.commands (module), 176

FobiFindBrokenEntries (module), 175

FobiFoundation5WidgetApps fobi_migrate_03_to_04 (module), 175

FobiFormElementsPlugins (module), 176

FobiIntegrationCommands fobi_update_plugin_data (module), 176

FobiTemplateTagsFobiTags widget apps fobi.settings (module), 221

FobiTemplatetagsFobiTags form elements fobi.templatetags.fobi_tags (module), 176

FobiFoundation5WidgetCompat (module), 177

FobiTests base (module), 178

FobiTests helpers (module), 178

FobiTests test_browser_build_dynamic_forms (module), 178

FobiTests test_core (module), 180

FobiTests test_form importers_mailchimp (module), 180

FobiUrls (module), 181

FobiUrls edit (module), 181

FobiUrls view (module), 181

FobiUtils (module), 222

FobiValidators (module), 223

FobiViews (module), 224

FobiWidgets (module), 227

FobibrowserBuldDynamicFormsTest (class in fobi.tests.test_browser_build_dynamic_forms), 179

FobicoreTest (class in fobi.tests.test_core), 180

FobiDataStructureTest (class in fobi.tests.test_sortable_dict), 180

FobidynamicFormsTest (class in fobi.tests.test_dynamic_forms), 180

FobiformWidget (class in fobi.contrib.apps.feincms_integration.widgets), 101

FobiformWidget.Meta (class in fobi.contrib.apps.feincms_integration.widgets), 101

form (fobi.admin.FormElementEntryInlineAdmin attribute), 182

form (fobi.admin.ModelFormHandlerEntryInlineAdmin attribute), 183

form (fobi.base.BasePlugin attribute), 186

form (fobi.contrib.plugins.form_elements.content.content_image.fobi_form attribute), 105

form (fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin.security.captcha.fobi_form_elements.attribute), 107	form (fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.ContentVideoPlugin.security.honeypot.fobi_form_elements.attribute), 109	form (fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.BooleanFields.SelectPlugin.security.recaptcha.fobi_form_elements.attribute), 110	form (fobi.contrib.plugins.form_elements.fields.checkbox_select.fobi_form_elements.CheckBoxSelectPlugin.security.recaptcha.fobi_form_elements.attribute), 111	form (fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.DateFieldPlugin.fobi_form_handlers.mail.fobi_form_handlers.MailHandler.attribute), 113	form (fobi.contrib.plugins.form_elements.fields.date_drop_down.felix (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme.attribute), 114	form (fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.DateTimeFieldPlugin.fobi_themes.admin_style_theme.fobi_themes.admin_style_theme.attribute), 115	form (fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.DecimalFieldPlugin.fobi_themes.foundation5.fobi_themes.foundation5_theme.attribute), 117	form (fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements.EmailInputPlugin.simple.fobi_themes.SimpleTheme.attribute), 118	form (fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.FileInputPlugin.fobi_themes.db_store.models.Sa(attribute), 119	form (fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements.FloatFieldPlugin_class(attribute), 121	form (fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements.HiddenInputPlugin(attribute), 122	form (fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements.TextInputPlugin.fobi_themes.djangocms_admin_style_theme.fobi_themes(attribute), 124	form (fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.IntegerInputPlugin_class(attribute), 125	form (fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.IPAddressInputPlugin(attribute), 126	form (fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.NullBooleanSelectPlugin.simple.fobi_themes(attribute), 126	form (fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements.PasswordInputPlugin.djangocms_admin_style_theme.fobi_themes(attribute), 127	form (fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements.RadioButtonPlugin.simple.fobi_themes.SimpleTheme(attribute), 129	form (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.RegexInputPlugin_class(attribute), 130	form (fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.SelectInputPlugin(attribute), 132	form (fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements.SelectModelObjectInputPlugin.fobi_themes(attribute), 133	form (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultipleInputPlugin(attribute), 136	form (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements.SelectMultipleModelObjectsInputPlugin(attribute), 138	form (fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin.simple.fobi_themes.SimpleTheme(attribute), 140	form (fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin(attribute), 141	form (fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin(attribute), 143	form (fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin.simple.fobi_themes.SimpleTheme(attribute), 145
--	---	---	---	--	--	---	---	--	--	---	---	--	---	--	--	---	---	---	---	---	--	--	--	---	---	--

form_element_checkbox_html_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164	form_entry_submitted_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173
form_element_checkbox_html_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 166	form_entry_submitted_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164
form_element_checkbox_html_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171	form_entry_submitted_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 167
form_element_checkbox_html_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173	form_entry_submitted_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171
form_element_html_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164	form_entry_submitted_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173
form_element_html_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 167	form_fieldset_entry (fobi.models.ModelFormEntry attribute), 219
form_element_html_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171	form_fieldset_entry_id (fobi.models.ModelFormEntry attribute), 219
form_element_html_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173	form_importer_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164
form_entry (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 101	form_importer_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164
form_entry (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry attribute), 153	form_importers() (in module fobi.context_processors), 160
form_entry (fobi.models.AbstractPluginEntry attribute), 214	form_list (fobi.contrib.plugins.form_importers.mailchimp_importer.views.MVC attribute), 161
form_entry (fobi.models.ModelFormEntry attribute), 219	form_list_container_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164
form_entry (fobi.models.ModelFormEntry attribute), 220	form_list_container_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 167
form_entry (fobi.models.FormHandlerEntry attribute), 221	form_list_container_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171
form_entry_id (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 102	form_list_container_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173
form_entry_id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry attribute), 153	form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164
form_entry_id (fobi.models.AbstractPluginEntry attribute), 214	form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171
form_entry_id (fobi.models.ModelFormEntry attribute), 220	form_properties_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 167
form_entry_submitted() (in module fobi.views), 226	form_properties_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 167
form_entry_submitted_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164	
form_entry_submitted_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme attribute), 167	
form_entry_submitted_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171	

form_properties_snippet_template_name	FormElement.MultipleObjectsReturned, 212
(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171	formfieldsetentry_set (fobi.compat.User attribute), 197
form_properties_snippet_template_name	FormElementAdmin (class in fobi.admin), 181
(fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173	FormElementEntry (class in fobi.models), 219
form_radio_element_html_class	FormElementEntry.DoesNotExist, 219
(fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167	FormElementEntry.MultipleObjectsReturned, 219
form_radio_element_html_class	formelemententry_set (fobi.models.FormEntry attribute), 217
(fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173	formelemententry_set (fobi.models.FormFieldsetEntry attribute), 220
form_sent_get_param (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 102	FormElementEntryAdmin (class in fobi.admin), 181
form_sent_get_param (fobi.integration.processors.IntegrationProcessor attribute), 175	FormElementEntryAdmin.Meta (class in fobi.admin), 181
form_snippet_template_name	FormElementEntryFormSet (in module fobi.forms), 206
(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164	FormElementEntryInlineAdmin (class in fobi.admin), 182
form_snippet_template_name	FormElementPlugin (class in fobi.base), 189
(fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167	FormElementPluginPluginDataStorage (class in fobi.base), 184
form_snippet_template_name	FormElementPluginDoesNotExist, 203
(fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167	FormElementPluginError, 204
form_snippet_template_name	FormElementPluginRegistry (class in fobi.base), 192
(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171	FormElementPluginWidget (class in fobi.base), 196
form_snippet_template_name	FormElementPluginWidgetRegistry (class in fobi.base), 196
(fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173	FormEntry (class in fobi.models), 216
form_submit_button_text	FormEntry.DoesNotExist, 216
(fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 102	FormEntry.MultipleObjectsReturned, 216
form_template_name (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 102	formentry_set (fobi.compat.User attribute), 197
form_title (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 102	formentry_set (fobi.models.FormWizardEntry attribute), 217
form_view_snippet_template_name	FormEntryAdmin (class in fobi.admin), 182
(fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167	FormEntryForm (class in fobi.forms), 205
form_view_snippet_template_name	FormEntryForm.Meta (class in fobi.forms), 205
(fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173	FormFieldPlugin (class in fobi.base), 195
form_wizard_entry (fobi.models.FormEntry attribute), 217	FormFieldsetEntry (class in fobi.models), 220
form_wizard_entry_id (fobi.models.FormEntry attribute), 217	FormFieldsetEntry.DoesNotExist, 220
form_wizard_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164	FormFieldsetEntry.MultipleObjectsReturned, 220
formatted_saved_data() (fobi.contrib.plugins.form_handlers.SaveFormHandler method), 153	formfieldsetentry_set (fobi.models.FormEntry attribute), 217
FormCallback (class in fobi.base), 191	FormFieldsetEntryAdmin (class in fobi.admin), 182
FormCallbackError, 204	FormFieldsetEntryAdmin.Meta (class in fobi.admin), 183
FormCallbackRegistry (class in fobi.base), 192	FormFieldsetEntryForm (class in fobi.forms), 206
FormElement (class in fobi.models), 212	FormFieldsetEntryForm.Meta (class in fobi.forms), 206
FormElement.DoesNotExist, 212	FormHandler (class in fobi.models), 213
	FormHandler.DoesNotExist, 213
	FormHandler.MultipleObjectsReturned, 213
	formhandlerentry_set (fobi.models.FormEntry attribute), 217
	FormHandlerAdmin (class in fobi.admin), 183
	FormHandlerEntry (class in fobi.models), 220
	FormHandlerEntry.DoesNotExist, 221
	FormHandlerEntry.MultipleObjectsReturned, 221
	formhandlerentry_set (fobi.models.FormEntry attribute), 217

[FormHandlerEntryAdmin](#) (class in [fobi.admin](#)), [183](#)
[FormHandlerEntryAdmin.Meta](#) (class in [fobi.admin](#)), [183](#)
[FormHandlerEntryForm](#) (class in [fobi.forms](#)), [207](#)
[FormHandlerEntryForm.Meta](#) (class in [fobi.forms](#)), [207](#)
[FormHandlerEntryInlineAdmin](#) (class in [fobi.admin](#)), [183](#)
[FormHandlerForm](#) (class in [fobi.forms](#)), [207](#)
[FormHandlerForm.Meta](#) (class in [fobi.forms](#)), [207](#)
[FormHandlerPlugin](#) (class in [fobi.base](#)), [190](#)
[FormHandlerPluginDataStorage](#) (class in [fobi.base](#)), [184](#)
[FormHandlerPluginDoesNotExist](#), [204](#)
[FormHandlerPluginError](#), [204](#)
[FormHandlerPluginRegistry](#) (class in [fobi.base](#)), [192](#)
[FormHandlerPluginWidget](#) (class in [fobi.base](#)), [196](#)
[FormHandlerPluginWidgetRegistry](#) (class in [fobi.base](#)), [195](#)
[FormImporterPluginRegistry](#) (class in [fobi.form_importers](#)), [205](#)
[FormPluginError](#), [204](#)
[forms_list_template](#) ([fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#) attribute), [164](#)
[forms_list_template](#) ([fobi.contrib.themes.djangocms_admin_style.fobi_themes.DjangoCMSAdminStyleTheme](#) attribute), [167](#)
[forms_list_template](#) ([fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#) attribute), [171](#)
[forms_list_template](#) ([fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#) attribute), [173](#)
[FormWizardEntry](#) (class in [fobi.models](#)), [215](#)
[FormWizardEntry.DoesNotExist](#), [215](#)
[FormWizardEntry.MultipleObjectsReturned](#), [215](#)
[formwizardentry_set](#) ([fobi.compat.User](#) attribute), [198](#)
[Foundation5Theme](#) (class in [fobi.contrib.themes.foundation5.fobi_themes](#)), [170](#)

G

[get\(\)](#) ([fobi.base.BaseRegistry](#) method), [191](#)
[get_absolute_url\(\)](#) ([fobi.models.FormEntry](#) method), [217](#)
[get_absolute_url\(\)](#) ([fobi.models.FormWizardEntry](#) method), [215](#)
[get_allowed_form_element_plugin_uids\(\)](#) (in module [fobi.utils](#)), [222](#)
[get_allowed_form_handler_plugin_uids\(\)](#) (in module [fobi.utils](#)), [223](#)
[get_allowed_plugin_uids\(\)](#) (in module [fobi.utils](#)), [222](#)
[get_app_label_and_model_name\(\)](#) (in module [fobi.helpers](#)), [210](#)
[get_callbacks\(\)](#) ([fobi.base.FormCallbackRegistry](#) method), [192](#)
[get_cloned_plugin_data\(\)](#) ([fobi.base.BasePlugin](#) method), [186](#)
[get_crop_filter\(\)](#) (in module [fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements](#)), [106](#)

[get_custom_actions\(\)](#) ([fobi.base.FormHandlerPlugin](#) method), [190](#)
[get_fobi_form_handler_plugin_custom_actions\(\)](#) (in module [fobi.templatetags.fobi_tags](#)), [176](#)
[get_fobi_plugin\(\)](#) (in module [fobi.templatetags.fobi_tags](#)), [176](#)
[get_form\(\)](#) ([fobi.base.BasePlugin](#) method), [187](#)
[get_form_data\(\)](#) ([fobi.form_importers.BaseFormImporter](#) method), [204](#)
[get_form_element_plugin_widget\(\)](#) (in module [fobi.base](#)), [196](#)
[get_form_field_instances\(\)](#) ([fobi.base.FormElementPlugin](#) method), [189](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements](#) method), [105](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements](#) method), [107](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements](#) method), [108](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements](#) method), [110](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements](#) method), [111](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements](#) method), [113](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements](#) method), [114](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements](#) method), [115](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements](#) method), [117](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements](#) method), [118](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements](#) method), [119](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements](#) method), [121](#)
[get_form_field_instances\(\)](#) ([fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements](#) method), [122](#)

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements.TextInputPlugin method), 124	get_form_field_instances() (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin method), 150
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.IntegerPlugin method), 125	get_form_field_type() (in module fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin), 177
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.IPAddressPlugin method), 126	get_form_handler_plugin_widget() (in module fobi.base), 196
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.IPAddressPlugin method), 126	get_form_handler_plugin_widget() (in module fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin), 177
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.NullBooleanSelectPlugin method), 126	get_form_importer_plugin_uids() (in module fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin), 177
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.NullBooleanSelectPlugin method), 126	get_form_importer_plugin_urls() (in module fobi.form_importers), 205
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements.PasswordInputPlugin method), 127	get_form_importers() (fobi.form_importers), 205
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements.PasswordInputPlugin method), 127	get_form_kwargs() (fobi.form_importers.mailchimp_importer.MailChimpFormImporter method), 161
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements.RadioInputPlugin method), 129	get_form_template_choices() (in module fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin), 100
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.RegexInputPlugin method), 130	get_form_template_choices() (in module fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin), 101
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.SelectInputPlugin method), 132	get_form_template_choices() (in module fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin), 103
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select_model_obj.fobi_form_elements.SelectModelObjectPlugin method), 133	get_form_template_name_display() (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin method), 102
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultipleModelObjectsPlugin method), 136	get_full_name() (fobi.helpers.StrippedUser method), 210
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultipleModelObjectsPlugin method), 136	get_full_name() (for fobi.helpers.StrippedUser), 210
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objs.fobi_form_elements.SelectMultipleModelObjectsPlugin method), 138	get_initialised_create_form() (fobi.base.BasePlugin method), 187
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objs.fobi_form_elements.SelectMultipleModelObjectsPlugin method), 138	get_initialised_create_form_or_404() (fobi.base.BasePlugin method), 187
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin method), 140	get_initialised_create_form_or_404() (fobi.base.BasePlugin method), 187
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin method), 140	get_initialised_edit_form_or_404() (fobi.base.BasePlugin method), 187
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin method), 141	get_instance() (fobi.base.BasePlugin method), 187
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin method), 143	get_next_by_created() (fobi.contrib.plugins.form_handlers.db_store.models.models.DbStoreModel method), 198
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin method), 143	get_next_by_date_joined() (fobi.compat.User method), 198
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin method), 145	get_or_create_db_model() (fobi.contrib.plugins.form_handlers.db_store.models.models.DbStoreModel method), 198
get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin method), 145	get_or_create_db_model() (fobi.contrib.plugins.form_handlers.db_store.models.models.DbStoreModel method), 198
get_form_field_instances() (fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaInputPlugin method), 146	get_or_create_db_model() (fobi.contrib.plugins.form_handlers.db_store.models.models.DbStoreModel method), 198
get_form_field_instances() (fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin method), 148	get_or_create_db_model() (fobi.contrib.plugins.form_handlers.db_store.models.models.DbStoreModel method), 198
get_form_field_instances() (fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements.ReCaptchaInputPlugin method), 149	get_plugin() (fobi.models.AbstractPluginEntry method), 204

get_plugin_data() (fobi.base.BasePluginForm method), 185	get_setting() (in module fobi.contrib.apps.djangocms_integration.conf), 100
get_plugin_form_data() (fobi.base.BasePlugin method), 187	get_setting() (in module fobi.contrib.apps.feincms_integration.conf), 101
get_plugin_widget() (in module fobi.base), 196	get_setting() (in module fobi.contrib.apps.mezzanine_integration.conf), 103
get_previous_by_created() (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry method), 153	get_setting() (in module fobi.contrib.plugins.form_elements.content.content_image.conf), 105
get_previous_by_date_joined() (fobi.compat.User method), 198	get_setting() (in module fobi.contrib.plugins.form_elements.content.content_video.conf), 109
get_queryset() (fobi.admin.BasePluginModelAdmin method), 181	get_setting() (in module fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf), 111
get_queryset() (fobi.admin.FormElementEntryAdmin method), 182	get_setting() (in module fobi.contrib.plugins.form_elements.fields.file.conf), 119
get_queryset() (fobi.admin.FormHandlerEntryAdmin method), 183	get_setting() (in module fobi.contrib.plugins.form_elements.fields.radio.conf), 128
get_queryset() (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormEntryAdmin method), 151	get_setting() (in module fobi.contrib.plugins.form_elements.fields.select.conf), 131
get_registered_form_callbacks() (in module fobi.base), 194	get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_model_object.conf), 133
get_registered_form_element_plugin_uids() (in module fobi.base), 193	get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf), 135
get_registered_form_element_plugins() (in module fobi.base), 193	get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_multiple.conf), 136
get_registered_form_handler_plugin_uids() (in module fobi.base), 194	get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.conf), 137
get_registered_form_handler_plugins() (in module fobi.base), 193	get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_object.conf), 139
get_registered_models() (in module fobi.helpers), 208	get_setting() (in module fobi.contrib.plugins.form_elements.security.honeypot.conf), 147
get_registered_plugin_uids() (in module fobi.base), 193	get_setting() (in module fobi.contrib.plugins.form_handlers.db_store.conf), 152
get_registered_plugins() (fobi.models.AbstractPluginEntry method), 215	get_setting() (in module fobi.contrib.plugins.form_handlers.mail.conf), 156
get_registered_plugins() (fobi.models.AbstractPluginModel method), 211	get_short_name() (fobi.helpers.StrippedUser method), 210
get_registered_plugins() (fobi.models.FormElement method), 212	
get_registered_plugins() (fobi.models.FormElementEntry method), 219	
get_registered_plugins() (fobi.models.FormHandler method), 213	
get_registered_plugins() (fobi.models.FormHandlerEntry method), 221	
get_registered_plugins() (in module fobi.base), 193	
get_registered_theme_uids() (in module fobi.base), 195	
get_registered_themes() (in module fobi.base), 195	
get_registry() (fobi.models.AbstractPluginEntry method), 215	
get_registry() (fobi.models.FormElementEntry method), 219	
get_registry() (fobi.models.FormHandlerEntry method), 221	
get_select_field_choices() (in module fobi.helpers), 209	
get_setting() (in module fobi.conf), 199	

[get_success_page_template_choices\(\) \(in module fobi.contrib.apps.djangocms_integration.helpers\), 100](#)
[get_success_page_template_choices\(\) \(in module fobi.contrib.apps.feincms_integration.helpers\), 101](#)
[get_success_page_template_choices\(\) \(in module fobi.contrib.apps.mezzanine_integration.helpers\), 103](#)
[get_success_page_template_name_display\(\) \(fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method\), 102](#)
[get_template_choices\(\) \(in module fobi.integration.helpers\), 174](#)
[get_template_names\(\) \(fobi.form_importers.BaseFormImporter method\), 204](#)
[get_updated_plugin_data\(\) \(fobi.base.BasePlugin method\), 187](#)
[get_urls\(\) \(fobi.admin.FormElementAdmin method\), 181](#)
[get_urls\(\) \(fobi.admin.FormHandlerAdmin method\), 183](#)
[get_user_form_element_plugins\(\) \(in module fobi.utils\), 222](#)
[get_user_form_element_plugins_grouped\(\) \(in module fobi.utils\), 223](#)
[get_user_form_handler_plugin_uids\(\) \(in module fobi.utils\), 223](#)
[get_user_form_handler_plugins\(\) \(in module fobi.utils\), 223](#)
[get_user_form_handler_plugins_grouped\(\) \(in module fobi.utils\), 223](#)
[get_user_plugin_uids\(\) \(in module fobi.utils\), 222](#)
[get_user_plugins\(\) \(in module fobi.utils\), 222](#)
[get_user_plugins_grouped\(\) \(in module fobi.utils\), 223](#)
[get_username\(\) \(fobi.helpers.StrippedUser method\), 210](#)
[get_widget\(\) \(fobi.base.BasePlugin method\), 187](#)
[get_wizard\(\) \(fobi.form_importers.BaseFormImporter method\), 204](#)
[graceful_export\(\) \(fobi.contrib.plugins.form_handlers.db_store.helpers.DatabaseExporter method\), 153](#)
[group \(fobi.base.BasePlugin attribute\), 188](#)
[group \(fobi.contrib.plugins.form_elements.content.content_image.ContentImagePlugin attribute\), 105](#)
[group \(fobi.contrib.plugins.form_elements.content.content_image.ContentImagePlugin attribute\), 107](#)
[group \(fobi.contrib.plugins.form_elements.content.content_image.ContentImagePlugin attribute\), 109](#)
[group \(fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.fields.boolean.FobiFormElementField attribute\), 110](#)
[group \(fobi.contrib.plugins.form_elements.fields.checkbox.select\(fobi.contrib.plugins.form_elements.checkbox.SelectFobiFormElementField attribute\), 111](#)
[group \(fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.fields.date.FobiFormElementField attribute\), 113](#)
[group \(fobi.contrib.plugins.form_elements.fields.date_drop_down.select\(fobi.contrib.plugins.form_elements.date_drop_down.SelectFobiFormElementField attribute\), 114](#)
[group \(fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.fields.datetime.FobiFormElementField attribute\), 115](#)
[group \(fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.fields.decimal.FobiFormElementField attribute\), 117](#)
[group \(fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements.fields.email.FobiFormElementField attribute\), 118](#)
[group \(fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.fields.file.FobiFormElementField attribute\), 119](#)
[group \(fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements.fields.float.FobiFormElementField attribute\), 121](#)
[group \(fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements.fields.hidden.FobiFormElementField attribute\), 122](#)
[group \(fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements.fields.input.FobiFormElementField attribute\), 124](#)
[group \(fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.fields.integer.FobiFormElementField attribute\), 125](#)
[group \(fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.fields.ip_address.FobiFormElementField attribute\), 126](#)
[group \(fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.fields.null_boolean.FobiFormElementField attribute\), 127](#)
[group \(fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements.fields.password.FobiFormElementField attribute\), 127](#)
[group \(fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements.fields.radio.FobiFormElementField attribute\), 129](#)
[group \(fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.fields.regex.FobiFormElementField attribute\), 130](#)
[group \(fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.fields.select.FobiFormElementField attribute\), 132](#)
[group \(fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements.fields.select_model_object.FobiFormElementField attribute\), 134](#)
[group \(fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.fields.select_multiple.FobiFormElementField attribute\), 136](#)
[group \(fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.fobi_form_elements.fields.select_multiple_model_object.FobiFormElementField attribute\), 138](#)
[group \(fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.fields.slug.FobiFormElementField attribute\), 140](#)
[group \(fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.fields.text.FobiFormElementField attribute\), 143](#)
[group \(fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.fields.time.FobiFormElementField attribute\), 143](#)
[group \(fobi.contrib.plugins.form_elements.url.fobi_form_elements.fields.url.FobiFormElementField attribute\), 145](#)
[group \(fobi.contrib.plugins.form_elements.validation.fobi_form_elements.validation.FobiFormElementField attribute\), 146](#)
[group \(fobi.contrib.plugins.form_elements.validation.fobi_form_elements.validation.FobiFormElementField attribute\), 148](#)
[group \(fobi.contrib.plugins.form_elements.validation.fobi_form_elements.validation.FobiFormElementField attribute\), 149](#)
[group \(fobi.contrib.plugins.form_elements.checkbox.select\(fobi.contrib.plugins.form_elements.checkbox.SelectFobiFormElementField attribute\), 150](#)
[group \(fobi.contrib.plugins.form_elements.date.fobi_form_elements.fields.date.FobiFormElementField attribute\), 198](#)
[groups \(fobi.models.AbstractPluginModel attribute\), 211](#)
[groups \(fobi.models.FormHandler attribute\), 213](#)

groups_list() (fobi.models.AbstractPluginModel method), 212

H

handle() (fobi.management.commands.fobi_find_broken_entries.Command method), 175

handle() (fobi.management.commands.fobi_migrate_03_to_04.Command method), 175

handle() (fobi.management.commands.fobi_sync_plugins.Command method), 176

handle() (fobi.management.commands.fobi_update_plugin_data.Command method), 176

handle_uploaded_file() (in module fobi.contrib.plugins.form_elements.content.content_image_helpers), 106

handle_uploaded_file() (in module fobi.helpers), 208

has_add_permission() (fobi.admin.BasePluginModelAdmin method), 181

has_edit_form_entry_permissions() (in module fobi.templatetags.fobi_tags), 177

has_value (fobi.base.FormElementPlugin attribute), 190

has_value (fobi.base.FormFieldPlugin attribute), 195

help_text (fobi.base.BaseFormFieldPluginForm attribute), 195

help_text (fobi.base.BasePlugin attribute), 188

HiddenInputForm (class in fobi.contrib.plugins.form_elements.fields.hidden.forms), 122

HiddenInputPlugin (class in fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements), 122

hide_form_title (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 102

hide_success_page_title (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 102

HoneypotField (class in fobi.contrib.plugins.form_elements.security.honeypot.fields), 147

HoneypotInputForm (class in fobi.contrib.plugins.form_elements.security.honeypot.forms), 148

HoneypotInputPlugin (class in fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements), 148

html_class (fobi.base.BasePlugin attribute), 188

html_classes (fobi.base.BasePlugin attribute), 188

html_id (fobi.base.BasePlugin attribute), 188

HTTPRepostForm (class in fobi.contrib.plugins.form_handlers.http_repost.forms), 155

HTTPRepostHandlerPlugin (class in fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers), 155

id (fobi.compat.User attribute), 198

id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormDataEntry attribute), 153

id (fobi.models.FormElement attribute), 213

id (fobi.models.FormElementEntry attribute), 219

id (fobi.models.FormEntry attribute), 217

id (fobi.models.FormFieldsetEntry attribute), 220

id (fobi.models.FormHandler attribute), 214

id (fobi.models.FormHandlerEntry attribute), 221

id (fobi.models.FormWizardEntry attribute), 215

import_data() (fobi.form_importers.BaseFormImporter method), 204

import_form_entry() (in module fobi.views), 226

import_form_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes attribute), 167

import_form_entry_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171

import_form_entry_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173

import_form_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes attribute), 167

import_form_entry_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171

import_form_entry_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173

ImportFormEntryForm (class in fobi.forms), 206

inproperty (fobi.forms.Widget), 201

inlines (fobi.admin.FormEntryAdmin attribute), 182

input_type (fobi.widgets.NumberInput attribute), 227

InputForm (class in fobi.contrib.plugins.form_elements.fields.input.forms), 124

InputPlugin (class in fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements), 124

insert() (fobi.data_structures.SortableDict method), 200

insert_after_key() (fobi.data_structures.SortableDict method), 200

insert_before_key() (fobi.data_structures.SortableDict method), 200

IntegerInputForm (class in fobi.contrib.plugins.form_elements.fields.integer.forms), 125

IntegerInputPlugin (class in fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements), 125

integration_processors() (fobi.integration.processors.IntegrationProcessor method), 175

IntegrationProcessor (class in fobi.integration.processors), 174

InvalidRegistryItemType, 203

IPAddressInputPlugin (class in fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config attribute), 126

is_ajax() (fobi.helpers.StrippedRequest method), 211

is_anonymous() (fobi.helpers.StrippedUser method), 210

is_cloneable (fobi.models.FormEntry attribute), 218

is_cloneable (fobi.models.FormWizardEntry attribute), 215

is_fobi_setup_completed() (in module fobi.tests.base), 178

is_hidden (fobi.base.FormElementPlugin attribute), 190

is_hidden (fobi.contrib.plugins.form_elements.fields.hidden_input.plugin attribute), 122

is_hidden (fobi.contrib.plugins.form_elements.security.honey_pot.plugin attribute), 148

is_hidden (fobi.contrib.plugins.form_handlers.mail.widgets.MultiFormWidget attribute), 159

is_public (fobi.models.FormEntry attribute), 218

is_public (fobi.models.FormWizardEntry attribute), 216

is_repeatable (fobi.models.FormFieldsetEntry attribute), 220

is_secure() (fobi.helpers.StrippedRequest method), 211

items() (fobi.data_structures.SortableDict method), 200

iterable_to_dict() (in module fobi.helpers), 208

iteritems() (fobi.data_structures.SortableDict method), 200

iterkeys() (fobi.data_structures.SortableDict method), 200

itervalues() (fobi.data_structures.SortableDict method), 200

J

js (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDataEntry attribute), 151

JSONDataExporter (class in fobi.helpers), 211

K

keys() (fobi.data_structures.SortableDict method), 200

L

label (fobi.apps.Config attribute), 184

label (fobi.base.BaseFormFieldPluginForm attribute), 195

label (fobi.contrib.apps.djangocms_integration.apps.Config attribute), 99

label (fobi.contrib.apps.feincms_integration.apps.Config attribute), 100

label (fobi.contrib.apps.mezzanine_integration.apps.Config attribute), 103

label (fobi.contrib.plugins.form_elements.content.content_image.apps.Config attribute), 105

label (fobi.contrib.plugins.form_elements.content.content_text.apps.Config attribute), 107

label (fobi.contrib.plugins.form_elements.content.content_video.apps.Config attribute), 108

label (fobi.contrib.plugins.form_elements.fields.boolean.apps.Config attribute), 110

label (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps.Config attribute), 111

label (fobi.contrib.plugins.form_elements.fields.date.apps.Config attribute), 113

label (fobi.contrib.plugins.form_elements.fields.date_drop_down.apps.Config attribute), 114

label (fobi.contrib.plugins.form_elements.fields.datetime.apps.Config attribute), 115

label (fobi.contrib.plugins.form_elements.fields.decimal.apps.Config attribute), 117

label (fobi.contrib.plugins.form_elements.fields.email.apps.Config attribute), 118

label (fobi.contrib.plugins.form_elements.fields.file.apps.Config attribute), 119

label (fobi.contrib.plugins.form_elements.fields.float.apps.Config attribute), 121

label (fobi.contrib.plugins.form_elements.fields.hidden.apps.Config attribute), 122

label (fobi.contrib.plugins.form_elements.fields.input.apps.Config attribute), 123

label (fobi.contrib.plugins.form_elements.fields.integer.apps.Config attribute), 125

label (fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config attribute), 126

label (fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config attribute), 126

label (fobi.contrib.plugins.form_elements.fields.password.apps.Config attribute), 127

label (fobi.contrib.plugins.form_elements.fields.radio.apps.Config attribute), 128

label (fobi.contrib.plugins.form_elements.fields.regex.apps.Config attribute), 130

label (fobi.contrib.plugins.form_elements.fields.select.apps.Config attribute), 131

label (fobi.contrib.plugins.form_elements.fields.select_model_object.apps.Config attribute), 133

label (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps.Config attribute), 134

label (fobi.contrib.plugins.form_elements.fields.select_multiple.apps.Config attribute), 135

label (fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.apps.Config attribute), 137

label (fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_object.apps.Config attribute), 139

label (fobi.contrib.plugins.form_elements.fields.slug.apps.Config attribute), 140

label (fobi.contrib.plugins.form_elements.fields.text.apps.Config attribute), 141

label (fobi.contrib.plugins.form_elements.fields.textarea.apps.Config display attribute),	142	(fobi.admin.ModelFormEntryAdmin attribute),	183
label (fobi.contrib.plugins.form_elements.fields.time.apps.Config display attribute),	143	(fobi.admin.FormHandlerEntryAdmin attribute),	183
label (fobi.contrib.plugins.form_elements.fields.url.apps.Config display attribute),	145	(fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDisplay attribute),	151
label (fobi.contrib.plugins.form_elements.security.captcha.apps.Config display attribute),	146	(fobi.admin.FormElementEntryAdmin attribute),	182
label (fobi.contrib.plugins.form_elements.security.honeypot.apps.Config display attribute),	147	(fobi.admin.FormEntryAdmin attribute),	182
label (fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config display attribute),	149	list_editable (fobi.admin.ModelFormEntryAdmin attribute),	183
label (fobi.contrib.plugins.form_elements.test.dummy.apps.Config display attribute),	150	list_filter (fobi.admin.FormElementEntryAdmin attribute),	182
label (fobi.contrib.plugins.form_handlers.db_store.apps.Config display attribute),	151	list_filter (fobi.admin.ModelFormEntryAdmin attribute),	183
label (fobi.contrib.plugins.form_handlers.http_repost.apps.Config display attribute),	155	list_filter (fobi.admin.FormHandlerEntryAdmin attribute),	183
label (fobi.contrib.plugins.form_handlers.mail.apps.Config display attribute),	156	list_filter (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDisplay attribute),	151
label (fobi.contrib.plugins.form_importers.mailchimp_importer.apps.Config display attribute),	159	lists_app_config (in module fobi.helpers),	208
label (fobi.contrib.themes.bootstrap3.apps.Config display attribute),	163	LIVE_SERVER_URL (fobi.tests.test_browser_build_dynamic_forms.FobiIView attribute),	179
label (fobi.contrib.themes.bootstrap3.widgets.form_element_identity_bootstrap3_widget.apps.Config display attribute),	161	load_plugin_data() (fobi.base.BasePlugin method),	188
label (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_foundation5_widget.apps.Config display attribute),	162	login_required_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Widget attribute),	198
label (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps.Config display attribute),	163	login_required_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Widget attribute),	175
M			
label (fobi.contrib.themes.djangocms_admin_style_theme.apps.Config display attribute),	166	MailChimpAPIKeyForm (class in fobi.contrib.plugins.form_importers.mailchimp_importer.forms),	160
label (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps.Config display attribute),	165	MailChimpImporter (class in fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_for),	159
label (fobi.contrib.themes.foundation5.apps.Config display attribute),	170	MailChimpListIDForm (class in fobi.contrib.plugins.form_importers.mailchimp_importer.forms),	160
label (fobi.contrib.themes.foundation5.widgets.form_element_mailchimp_importer_views_view.apps.Config display attribute),	168	map_field_name_to_label() (in module fobi.helpers),	208
label (fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps.Config display attribute),	168	mark_fobi_setup_as_completed() (in module fobi.tests.base),	178
label (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps.Config display attribute),	169	master_base_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Widget attribute),	164
label (fobi.contrib.themes.foundation5.widgets.form_handlers.mail_form.apps.Config display attribute),	169	master_base_template (fobi.contrib.themes.djangocms_admin_style_theme.Widget attribute),	167
label (fobi.contrib.themes.simple.apps.Config display attribute),	172		
label (fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps.Config display attribute),	172		
list_display (fobi.admin.BasePluginModelAdmin attribute),	181		
list_display (fobi.admin.FormElementEntryAdmin attribute),	182		
list_display (fobi.admin.FormEntryAdmin attribute),	182		

master_base_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171	media (fobi.admin.BasePluginModelAdmin attribute), 181	media (fobi.contrib.plugins.form_elements.fields.select_forms.SelectInputForm attribute), 133
master_base_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 173	media (fobi.admin.FormElementAdmin attribute), 181	media (fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectForm attribute), 134
media (fobi.admin.BasePluginModelAdmin attribute), 181	media (fobi.admin.FormElementEntryAdmin attribute), 182	media (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMpttModelObjectForm attribute), 135
media (fobi.admin.FormElementAdmin attribute), 181	media (fobi.admin.FormElementEntryAdmin attribute), 182	media (fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleForm attribute), 137
media (fobi.admin.FormElementEntryAdmin attribute), 182	media (fobi.admin.FormElementEntryInlineAdmin attribute), 182	media (fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.forms.SelectMultipleModelObjectForm attribute), 138
media (fobi.admin.FormEntryAdmin attribute), 182	media (fobi.admin.FormFieldsetEntryAdmin attribute), 183	media (fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_object.forms.SelectMultipleMpttModelObjectForm attribute), 139
media (fobi.admin.FormFieldsetEntryAdmin attribute), 183	media (fobi.admin.FormHandlerAdmin attribute), 183	media (fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm attribute), 141
media (fobi.admin.FormHandlerEntryAdmin attribute), 183	media (fobi.admin.FormHandlerEntryInlineAdmin attribute), 183	media (fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm attribute), 142
media (fobi.admin.FormHandlerEntryInlineAdmin attribute), 183	media (fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm attribute), 106	media (fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm attribute), 143
media (fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm attribute), 106	media (fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm attribute), 108	media (fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm attribute), 144
media (fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm attribute), 108	media (fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm attribute), 110	media (fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm attribute), 145
media (fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm attribute), 110	media (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleInputForm attribute), 112	media (fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaForm attribute), 146
media (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleInputForm attribute), 112	media (fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm attribute), 114	media (fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotForm attribute), 147
media (fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm attribute), 114	media (fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateDropDownInputForm attribute), 115	media (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDataForm attribute), 148
media (fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateDropDownInputForm attribute), 115	media (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm attribute), 116	media (fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm attribute), 149
media (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm attribute), 116	media (fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm attribute), 117	media (fobi.contrib.plugins.form_handlers.mail.forms.MailForm attribute), 150
media (fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm attribute), 117	media (fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm attribute), 119	media (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget attribute), 151
media (fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm attribute), 119	media (fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm attribute), 120	media (fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpImporterForm attribute), 152
media (fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm attribute), 120	media (fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm attribute), 121	media (fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpImporterForm attribute), 152
media (fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm attribute), 121	media (fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm attribute), 122	media (fobi.forms.BulkChangeFormElementPluginsForm attribute), 206
media (fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm attribute), 122	media (fobi.contrib.plugins.form_elements.fields.input.forms.InputForm attribute), 124	media (fobi.forms.BulkChangeFormHandlerPluginsForm attribute), 206
media (fobi.contrib.plugins.form_elements.fields.input.forms.InputForm attribute), 124	media (fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm attribute), 125	media (fobi.forms.FormEntryForm attribute), 206
media (fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm attribute), 125	media (fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm attribute), 128	media (fobi.forms.FormFieldsetEntryForm attribute), 206
media (fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm attribute), 128	media (fobi.contrib.plugins.form_elements.fields.radio.forms.RadioButtonForm attribute), 130	media (fobi.forms.FormHandlerEntryForm attribute), 207
media (fobi.contrib.plugins.form_elements.fields.radio.forms.RadioButtonForm attribute), 130	media (fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm attribute), 131	media (fobi.forms.ImportFormEntryForm attribute), 207
media (fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm attribute), 131		media (fobi.widgets.NumberInput attribute), 227
		media_css (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164

media_css (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 162

media_css (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 162

media_css (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 163

media_css (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167

media_css (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171

media_css (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 168

media_css (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 168

media_css (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 169

media_css (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 174

media_js (fobi.base.BasePlugin attribute), 188

media_js (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164

media_js (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 162

media_js (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 162

media_js (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DummyPluginWidget attribute), 163

media_js (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167

media_js (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171

media_js (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 168

media_js (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 168

media_js (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements.DatePluginWidget attribute), 169

media_js (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 174

message (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField attribute), 157

messages_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164

messages_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167

messages_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171

messages_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 174

META (fobi.helpers.StrippedRequest attribute), 210

name (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements.
attribute), 111

name (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements.
attribute), 111

name (fobi.contrib.plugins.form_elements.fields.date.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 113

name (fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.fobi_form_elements.
attribute), 113

name (fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.fobi_form_elements.
attribute), 114

name (fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.fobi_form_elements.
attribute), 114

name (fobi.contrib.plugins.form_elements.fields.datetime.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 115

name (fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements.fobi_form_elements.
attribute), 116

name (fobi.contrib.plugins.form_elements.fields.decimal.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 117

name (fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.fobi_form_elements.
attribute), 117

name (fobi.contrib.plugins.form_elements.fields.email.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 118

name (fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements.fobi_form_elements.
attribute), 118

name (fobi.contrib.plugins.form_elements.fields.file.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 119

name (fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.fobi_form_elements.
attribute), 119

name (fobi.contrib.plugins.form_elements.fields.float.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 121

name (fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements.fobi_form_elements.
attribute), 121

name (fobi.contrib.plugins.form_elements.fields.hidden.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 122

name (fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements.fobi_form_elements.
attribute), 122

name (fobi.contrib.plugins.form_elements.fields.input.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 123

name (fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements.fobi_form_elements.
attribute), 124

name (fobi.contrib.plugins.form_elements.fields.integer.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 125

name (fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.fobi_form_elements.
attribute), 125

name (fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 126

name (fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.fobi_form_elements.
attribute), 126

name (fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 126

name (fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.fobi_form_elements.
attribute), 127

name (fobi.contrib.plugins.form_elements.fields.password.apps.Config (fobi.contrib.plugins.form_elements.
attribute), 127

Index 273

post_processor() (fobi.contrib.plugins.form_elements.content.content_video_plugin method), 109

post_processor() (fobi.contrib.plugins.form_elements.test.dummy_plugin method), 150

pre_processor() (fobi.base.BasePlugin method), 188

prep_value() (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget method), 159

print_info() (in module fobi.tests.base), 178

process() (fobi.base.BasePlugin method), 189

process() (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method), 102

process_plugin_data() (fobi.base.BasePlugin method), 189

render() (fobi.contrib.plugins.form_elements.content.content_video_plugin method), 109

render() (fobi.contrib.plugins.form_elements.test.dummy_plugin method), 150

render() (fobi.base.BasePlugin method), 189

render() (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget method), 159

render() (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreFormHandler method), 152

run() (fobi.base.BaseFormHandlerPlugin method), 190

run() (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostFormHandler method), 155

run() (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailHandler method), 157

run_form_handlers() (in module fobi.base), 194

R

radio_fields (fobi.admin.FormEntryAdmin attribute), 182

RadioInputForm (class in fobi.contrib.plugins.form_elements.fields.radio.forms), 129

RadioInputPlugin (class in fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements), 129

readonly_fields (fobi.admin.BasePluginModelAdmin attribute), 181

readonly_fields (fobi.admin.FormElementEntryAdmin attribute), 182

readonly_fields (fobi.admin.FormEntryAdmin attribute), 182

readonly_fields (fobi.admin.FormHandlerEntryAdmin attribute), 183

readonly_fields (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormEntryAdmin attribute), 151

ReCaptchaInputForm (class in fobi.contrib.plugins.form_elements.security.recaptcha.forms), 149

ReCaptchaInputPlugin (class in fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements), 149

RegexInputForm (class in fobi.contrib.plugins.form_elements.fields.regex.forms), 130

RegexInputPlugin (class in fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements), 130

register() (fobi.base.BaseRegistry method), 192

register() (fobi.base.FormCallbackRegistry method), 192

registrationprofile (fobi.compat.User attribute), 198

render() (fobi.base.BasePlugin method), 189

render() (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method), 102

render() (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget method), 159

render_auth_link() (in module fobi templatetags.fobi_tags), 177

S

safe_text() (in module fobi.helpers), 209

save_plugin_data() (fobi.base.BasePluginForm method), 185

save_plugin_data() (fobi.contrib.plugins.form_elements.content.content_image_plugin method), 106

saved_data (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry attribute), 154

SavedFormEntry (class in fobi.contrib.plugins.form_handlers.db_store.models), 153

SavedFormEntry.DoesNotExist, 153

SavedFormEntry.MultipleObjectsReturned, 153

savedformdataentry_set (fobi.compat.User attribute), 198

savedformdataentry_set (fobi.models.FormEntry attribute), 218

SavedFormEntryAdmin (class in fobi.contrib.plugins.form_handlers.db_store.admin), 151

SavedFormEntryAdmin.Media (class in fobi.contrib.plugins.form_handlers.db_store.admin), 151

SavedFormEntryAdmin.Meta (class in fobi.contrib.plugins.form_handlers.db_store.admin), 151

SelectInputForm (class in fobi.contrib.plugins.form_elements.fields.select.forms), 132

SelectInputPlugin (class in fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements), 132

SelectModelObjectInputForm (class in fobi.contrib.plugins.form_elements.fields.select_model_object.forms), 134

SelectModelObjectInputPlugin (class in fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements), 133

SelectMPTTModelObjectInputForm (class in fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms), 135

[SelectMultipleInputForm](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_element](#)
[136](#) [method\(\)](#), [111](#)
[SelectMultipleInputPlugin](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_element](#)
[136](#) [method\(\)](#), [113](#)
[SelectMultipleModelObjectsInputForm](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_element](#)
[138](#) [method\(\)](#), [116](#)
[SelectMultipleModelObjectsInputPlugin](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_element](#)
[138](#) [method\(\)](#), [120](#)
[SelectMultipleMPTTModelObjectsInputForm](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_element](#)
[139](#) [method\(\)](#), [129](#)
[send_mail\(\)](#) (in module [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_handlers.mail.helpers](#)), ([fobi.contrib.plugins.form_elements.fields.select.fobi_form_element](#)
[158](#) [method\(\)](#), [132](#)
[setdefault\(\)](#) ([fobi.data_structures.SortableDict](#) method), [submit_plugin_form_data\(\)](#)
[201](#) ([fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_element](#)
[method\(\)](#), [134](#)
[setUp\(\)](#) ([fobi.tests.test_core.FobiCoreTest](#) method), [180](#) [method\(\)](#), [134](#)
[setUp\(\)](#) ([fobi.tests.test_dynamic_forms.FobiDynamicFormsTest](#) method), [submit_plugin_form_data\(\)](#)
[method\(\)](#), [180](#) ([fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_element](#)
[method\(\)](#), [136](#)
[setUp\(\)](#) ([fobi.tests.test_sortable_dict.FobiDataStructuresTest](#) method), [180](#) [submit_plugin_form_data\(\)](#)
[method\(\)](#), [180](#) [method\(\)](#), [138](#)
[setUpClass\(\)](#) ([fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsForTest](#)
[class method\(\)](#), [179](#) [method\(\)](#), [138](#)
[SimpleTheme](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.themes.simple.fobi_themes](#)), ([fobi.contrib.plugins.form_elements.fields.time.fobi_form_element](#)
[173](#) [method\(\)](#), [144](#)
[skip\(\)](#) (in module [fobi.tests.base](#)), [178](#) [success_page_message](#) ([fobi.models.FormEntry](#) attribute), [218](#)
[slug](#) ([fobi.models.FormEntry](#) attribute), [218](#) [success_page_template_name](#)
[slug](#) ([fobi.models.FormWizardEntry](#) attribute), [216](#) ([fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget](#)
[fobi.contrib.plugins.form_elements.fields.slug.forms](#)), attribute), [102](#)
[SlugInputForm](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_elements.fields.slug.forms](#)), ([fobi.contrib.apps.feincms_integration.widgets.FobiForm](#)
[140](#) [success_page_text](#) ([fobi.contrib.apps.feincms_integration.widgets.FobiForm](#)
[attribute\(\)](#), [102](#)
[SlugInputPlugin](#) (class in [submit_plugin_form_data\(\)](#)
[fobi.contrib.plugins.form_elements.fields.slug.fobi_form_element](#)), ([fobi.contrib.apps.feincms_integration.widgets.FobiForm](#)
[140](#) [success_page_title](#) ([fobi.contrib.apps.feincms_integration.widgets.FobiForm](#)
[attribute\(\)](#), [102](#)
[SortableDict](#) (class in [fobi.data_structures](#)), [200](#) [success_page_title](#) ([fobi.models.FormEntry](#) attribute),
[stage](#) ([fobi.base.FormCallback](#) attribute), [191](#) [218](#)
[storage](#) ([fobi.base.BasePlugin](#) attribute), [189](#) [sync_plugins\(\)](#) (in module [fobi.utils](#)), [222](#)
[storage](#) ([fobi.base.FormElementPlugin](#) attribute), [190](#) **T**
[storage](#) ([fobi.base.FormElementPluginWidget](#) attribute), [196](#) [tearDown\(\)](#) ([fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsForTest](#)
[storage](#) ([fobi.base.FormHandlerPlugin](#) attribute), [191](#) [method\(\)](#), [179](#)
[storage](#) ([fobi.base.FormHandlerPluginWidget](#) attribute), [196](#) [tearDownClass\(\)](#) ([fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsForTest](#)
[class method\(\)](#), [179](#)
[StrippedRequest](#) (class in [fobi.helpers](#)), [210](#) [templates](#) ([fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_element](#)
[attribute\(\)](#), [160](#)
[StrippedUser](#) (class in [fobi.helpers](#)), [210](#) [templates](#) ([fobi.form_importers.BaseFormImporter](#) attribute), [204](#)
[submit_plugin_form_data\(\)](#) ([fobi.base.FormElementPlugin](#) method), [190](#) [test_01_assemble_form_class_and_render_form\(\)](#)
[190](#) ([fobi.tests.test_dynamic_forms.FobiDynamicFormsTest](#)

method), 180	TextInputForm (class in fobi.contrib.plugins.form_elements.fields.text.forms), 141
test_01_get_registered_form_element_plugins() (fobi.tests.test_core.FobiCoreTest method), 180	TextInputPlugin (class in fobi.contrib.plugins.form_elements.fields.text.fobi_form_element), 141
test_01_sortable_dict_move_before_key() (fobi.tests.test_sortable_dict.FobiDataStructuresTest method), 180	theme() (in module fobi.context_processors), 199
test_02_get_registered_form_handler_plugins() (fobi.tests.test_core.FobiCoreTest method), 180	theme_uid (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3.datetime.datetime attribute), 162
test_02_sortable_dict_move_after_key() (fobi.tests.test_sortable_dict.FobiDataStructuresTest method), 180	theme_uid (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3.datetime attribute), 162
test_03_get_registered_form_callbacks() (fobi.tests.test_core.FobiCoreTest method), 180	theme_uid (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3.datetime attribute), 163
test_04_get_registered_themes() (fobi.tests.test_core.FobiCoreTest method), 180	theme_uid (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.date_bootstrap3.datetime attribute), 165
test_05_action_url() (fobi.tests.test_core.FobiCoreTest method), 180	theme_uid (fobi.contrib.themes.foundation5.widgets.form_elements.date_bootstrap3.datetime attribute), 168
test_1001_open_dashboard() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	theme_uid (fobi.contrib.themes.foundation5.widgets.form_elements.datetime_bootstrap3.datetime attribute), 168
test_2001_add_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	theme_uid (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_bootstrap3.datetime attribute), 169
test_2002_edit_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	theme_uid (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store.fobi_form_elements.datetime attribute), 170
test_2003_delete_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	ThemeDoesNotExist, 203
test_2004_submit_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	TimeInputForm (class in fobi.contrib.plugins.form_elements.fields.time.forms), 141
test_3001_add_form_elements() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	TimeInputPlugin (class in fobi.contrib.plugins.form_elements.fields.time.fobi_form_element), 141
test_3002_remove_form_elements() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	to_python() (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField method), 157
test_3003_edit_form_elements() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	utils (fobi.contrib.plugins.form_handlers.mail.fields.fobi.helpers), 208
test_4001_add_form_handlers() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	type (fobi.base.BaseRegistry attribute), 192
test_4002_remove_form_handlers() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 179	type (fobi.base.FormElementPluginRegistry attribute), 192
test_4003_edit_form_handlers() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 180	type (fobi.base.FormElementPluginWidgetRegistry attribute), 195
TextareaForm (class in fobi.contrib.plugins.form_elements.fields.textarea.forms), 142	type (fobi.base.FormHandlerPluginWidgetRegistry attribute), 192
	type (fobi.form_importers.FormImporterPluginRegistry attribute), 205
	uid (fobi.base.BasePlugin attribute), 189
	uid (fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element attribute), 105
	uid (fobi.contrib.plugins.form_elements.content.content_text.fobi_form_element attribute), 108

- uid (fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.security.honeypot.fobi_form_elements attribute), 109
- uid (fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.security.recaptcha.fobi_form_elements attribute), 110
- uid (fobi.contrib.plugins.form_elements.fields.checkbox_select.fobi_form_elements.Checktest.SelectMultipleInputPlugins.D attribute), 112
- uid (fobi.contrib.plugins.form_elements.fields.date.fobi_form_handlers.db_store.fobi_form_handlers.DBStore attribute), 113
- uid (fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_handlers.HTTP attribute), 114
- uid (fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_handlers.mail.fobi_form_handlers.MailHandler attribute), 116
- uid (fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_handlers.mailchimp_importers.mailchimp_importer.fobi_form_in attribute), 117
- uid (fobi.contrib.plugins.form_elements.fields.email.fobi_form_handlers.BaseFormImporter attribute), 118
- uid (fobi.contrib.plugins.form_elements.fields.file.fobi_form_handlers.djangoCMS_admin_style_theme.fobi_themes.Django attribute), 120
- uid (fobi.contrib.plugins.form_elements.fields.float.fobi_form_handlers.Foundation5.fobi_themes.Foundation5Theme attribute), 121
- uid (fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_handlers.SimpleTheme attribute), 122
- uid (fobi.contrib.plugins.form_elements.fields.input.fobi_form_handlers.BaseFormImporter attribute), 124
- uid (fobi.contrib.plugins.form_elements.fields.integer.fobi_form_handlers.Registry method), 125
- uid (fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_handlers.BaseFormImporter attribute), 126
- uid (fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_handlers.BaseFormImporter attribute), 127
- uid (fobi.contrib.plugins.form_elements.fields.password.fobi_form_handlers.BaseFormImporter attribute), 127
- uid (fobi.contrib.plugins.form_elements.fields.radio.fobi_form_handlers.BaseFormImporter attribute), 129
- uid (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_handlers.BaseFormImporter attribute), 130
- uid (fobi.contrib.plugins.form_elements.fields.select.fobi_form_handlers.BaseFormImporter attribute), 132
- uid (fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_handlers.BaseFormImporter attribute), 134
- uid (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_handlers.BaseFormImporter attribute), 136
- uid (fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_handlers.BaseFormImporter attribute), 138
- uid (fobi.contrib.plugins.form_elements.fields.slug.fobi_form_handlers.BaseFormImporter attribute), 140
- uid (fobi.contrib.plugins.form_elements.fields.text.fobi_form_handlers.BaseFormImporter attribute), 141
- uid (fobi.contrib.plugins.form_elements.fields.time.fobi_form_handlers.BaseFormImporter attribute), 144
- uid (fobi.contrib.plugins.form_elements.fields.url.fobi_form_handlers.BaseFormImporter attribute), 145
- uid (fobi.contrib.plugins.form_elements.security.captcha.fobi_form_handlers.BaseFormImporter attribute), 146
- uid (fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements attribute), 148
- uid (fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements attribute), 149
- uid (fobi.contrib.plugins.form_elements.Checktest.SelectMultipleInputPlugins.D attribute), 150
- uid (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStore attribute), 152
- uid (fobi.contrib.plugins.form_handlers.HTTP attribute), 155
- uid (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailHandler attribute), 158
- uid (fobi.contrib.plugins.form_handlers.mailchimp_importers.mailchimp_importer.fobi_form_in attribute), 160
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 164
- uid (fobi.contrib.plugins.form_handlers.djangoCMS_admin_style_theme.fobi_themes.Django attribute), 167
- uid (fobi.contrib.plugins.form_handlers.Foundation5.fobi_themes.Foundation5Theme attribute), 171
- uid (fobi.contrib.plugins.form_handlers.SimpleTheme attribute), 174
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 205
- uid (fobi.contrib.plugins.form_handlers.Registry method), 192
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 192
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 201
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 189
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 209
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 218
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 223
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 145
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 197
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 154
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 216
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 197
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 218
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 216
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 212
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 213
- uid (fobi.contrib.plugins.form_handlers.BaseFormImporter attribute), 214

users_list() (fobi.models.AbstractPluginModel method), 212

V

validate() (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField method), 157

validate_form_element_plugin_uid() (in module fobi.base), 193

validate_form_handler_plugin_uid() (in module fobi.base), 194

validate_initial_for_choices() (in module fobi.helpers), 209

validate_initial_for_multiple_choices() (in module fobi.helpers), 209

validate_plugin_data() (fobi.base.BaseFormFieldPluginForm method), 195

validate_plugin_data() (fobi.base.BasePluginForm method), 185

validate_submit_value_as() (in module fobi.helpers), 210

validate_theme_uid() (in module fobi.base), 195

value_for_index() (fobi.data_structures.SortableDict method), 201

values() (fobi.data_structures.SortableDict method), 201

view_embed_form_entry_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164

view_entries_icon_class (fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget attribute), 154

view_entries_icon_class (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements.DbStorePluginWidget attribute), 165

view_entries_icon_class (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_elements.DbStorePluginWidget attribute), 170

view_entries_icon_class (fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements.DbStorePluginWidget attribute), 172

view_form_entry() (in module fobi.views), 226

view_form_entry_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164

view_form_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167

view_form_entry_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171

view_form_entry_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 174

view_form_entry_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 164

view_form_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 167

view_form_entry_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 171

view_form_entry_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 174

view_saved_form_data_entries() (in module fobi.contrib.plugins.form_handlers.db_store.views), 154

view_saved_form_data_entries_template_name (fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget attribute), 154

view_saved_form_data_entries_template_name (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_elements.DbStorePluginWidget attribute), 170

W

widget (fobi.base.BasePlugin attribute), 189

widget (fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotField attribute), 148

widget (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField attribute), 157

wizard (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.BaseFormImporter attribute), 205