
django-fobi Documentation

Release 0.17.1

Artur Barseghyan <artur.barseghyan@gmail.com>

Jun 24, 2022

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

| | | |
|-----------|---|-----------|
| 11.1.3 | path/to/sample_importer/views.py | 33 |
| 11.1.4 | Form importer plugin final steps | 35 |
| 12 | Creating a form callback | 37 |
| 13 | Suggestions | 39 |
| 13.1 | Custom action for the form | 39 |
| 13.2 | When you want to customise too many things | 39 |
| 14 | Theming | 41 |
| 14.1 | Create a new theme | 41 |
| 14.2 | Make changes to an existing theme | 45 |
| 14.2.1 | Directory structure | 45 |
| 14.2.2 | fobi_themes.py | 45 |
| 14.2.3 | templates/override_simple_theme/base_view.html | 46 |
| 14.2.4 | templates/override_simple_theme/snippets/form_ajax.html | 46 |
| 15 | Form wizards | 47 |
| 15.1 | Basics | 47 |
| 15.2 | Bundled form wizard handler plugins | 47 |
| 16 | Integration with third-party apps and frameworks | 49 |
| 16.1 | Sample <i>IntegrationFormElementPlugin</i> | 49 |
| 16.1.1 | base.py | 49 |
| 16.1.2 | fobi_integration_form_elements.py | 50 |
| 16.2 | Sample <i>IntegrationFormHandlerPlugin</i> | 50 |
| 16.2.1 | base.py | 51 |
| 16.2.2 | fobi_integration_form_handlers.py | 52 |
| 17 | Permissions | 53 |
| 18 | Management commands | 55 |
| 19 | Tuning | 57 |
| 20 | Bundled plugins and themes | 59 |
| 20.1 | Bundled form element plugins | 59 |
| 20.1.1 | Fields | 59 |
| 20.1.2 | Content/presentation | 60 |
| 20.1.3 | Security | 60 |
| 20.1.4 | MPTT fields | 60 |
| 20.1.5 | Test | 61 |
| 20.2 | Bundled form handler plugins | 61 |
| 20.3 | Bundled themes | 61 |
| 21 | Third-party plugins and themes | 63 |
| 22 | HTML5 fields | 65 |
| 23 | Loading initial data using GET arguments | 67 |
| 24 | Dynamic initial values | 69 |
| 25 | Submitted form element plugins values | 71 |

| | | |
|------------|--|------------|
| 26 | Rendering forms using third-party libraries | 73 |
| 26.1 | Using <i>django-crispy-forms</i> | 73 |
| 26.2 | Using <i>django-floppyforms</i> | 73 |
| 27 | Import/export forms | 75 |
| 28 | Translations | 77 |
| 28.1 | Available translations | 77 |
| 28.2 | Overriding translations | 77 |
| 29 | Debugging | 79 |
| 30 | Testing | 81 |
| 30.1 | Browser tests | 81 |
| 30.1.1 | Set up ChromeDriver | 82 |
| 30.1.2 | Set up Firefox 47 | 82 |
| 30.1.3 | Set up headless Firefox | 82 |
| 30.1.4 | Setup PhantomJS | 83 |
| 31 | Writing documentation | 85 |
| 32 | Troubleshooting | 87 |
| 33 | Contributing | 89 |
| 34 | License | 91 |
| 35 | Support | 93 |
| 36 | Author | 95 |
| 37 | Screenshots | 97 |
| 37.1 | Bootstrap3 theme | 98 |
| 37.1.1 | Dashboard | 98 |
| 37.1.2 | Create a form | 99 |
| 37.1.3 | View/edit form | 100 |
| 37.1.3.1 | Form elements | 100 |
| 37.1.3.2 | Form handlers | 104 |
| 37.2 | Simple theme | 115 |
| 37.2.1 | View/edit form | 115 |
| 38 | Documentation | 121 |
| 38.1 | Quick start | 121 |
| 38.1.1 | Part 1: standard Django installation | 121 |
| 38.1.1.1 | Installation and configuration | 121 |
| 38.1.1.1.1 | Install the package in your environment. | 121 |
| 38.1.1.1.2 | INSTALLED_APPS | 121 |
| 38.1.1.1.3 | TEMPLATE_CONTEXT_PROCESSORS | 124 |
| 38.1.1.1.4 | urlpatterns | 124 |
| 38.1.1.1.5 | Update the database | 125 |
| 38.1.1.1.6 | Specify the active theme | 125 |
| 38.1.1.1.7 | Permissions | 125 |
| 38.1.2 | Part 2: Integration with DjangoCMS | 125 |
| 38.2 | Release history and notes | 125 |
| 38.2.1 | 0.17.1 | 126 |
| 38.2.2 | 0.17 | 126 |

| | | |
|---------|---------|-----|
| 38.2.3 | 0.16.4 | 126 |
| 38.2.4 | 0.16.3 | 126 |
| 38.2.5 | 0.16.2 | 126 |
| 38.2.6 | 0.16.1 | 126 |
| 38.2.7 | 0.16 | 127 |
| 38.2.8 | 0.15 | 127 |
| 38.2.9 | 0.14 | 127 |
| 38.2.10 | 0.13.10 | 127 |
| 38.2.11 | 0.13.9 | 128 |
| 38.2.12 | 0.13.8 | 128 |
| 38.2.13 | 0.13.7 | 128 |
| 38.2.14 | 0.13.6 | 128 |
| 38.2.15 | 0.13.5 | 128 |
| 38.2.16 | 0.13.4 | 128 |
| 38.2.17 | 0.13.3 | 128 |
| 38.2.18 | 0.13.2 | 129 |
| 38.2.19 | 0.13.1 | 129 |
| 38.2.20 | 0.13 | 129 |
| 38.2.21 | 0.12.20 | 129 |
| 38.2.22 | 0.12.19 | 129 |
| 38.2.23 | 0.12.18 | 129 |
| 38.2.24 | 0.12.17 | 130 |
| 38.2.25 | 0.12.16 | 130 |
| 38.2.26 | 0.12.15 | 130 |
| 38.2.27 | 0.12.14 | 131 |
| 38.2.28 | 0.12.13 | 131 |
| 38.2.29 | 0.12.12 | 131 |
| 38.2.30 | 0.12.11 | 132 |
| 38.2.31 | 0.12.10 | 132 |
| 38.2.32 | 0.12.9 | 132 |
| 38.2.33 | 0.12.8 | 132 |
| 38.2.34 | 0.12.7 | 132 |
| 38.2.35 | 0.12.6 | 132 |
| 38.2.36 | 0.12.5 | 133 |
| 38.2.37 | 0.12.4 | 133 |
| 38.2.38 | 0.12.3 | 133 |
| 38.2.39 | 0.12.2 | 133 |
| 38.2.40 | 0.12.1 | 133 |
| 38.2.41 | 0.12 | 133 |
| 38.2.42 | 0.11.13 | 133 |
| 38.2.43 | 0.11.12 | 134 |
| 38.2.44 | 0.11.11 | 134 |
| 38.2.45 | 0.11.10 | 134 |
| 38.2.46 | 0.11.9 | 134 |
| 38.2.47 | 0.11.8 | 134 |
| 38.2.48 | 0.11.7 | 134 |
| 38.2.49 | 0.11.6 | 135 |
| 38.2.50 | 0.11.5 | 135 |
| 38.2.51 | 0.11.4 | 135 |
| 38.2.52 | 0.11.3 | 135 |
| 38.2.53 | 0.11.2 | 135 |
| 38.2.54 | 0.11.1 | 135 |
| 38.2.55 | 0.11 | 135 |
| 38.2.56 | 0.10.7 | 136 |

| | | |
|-------------|---|-----|
| 38.2.57 | 0.10.6 | 136 |
| 38.2.58 | 0.10.5 | 136 |
| 38.2.59 | 0.10.4 | 136 |
| 38.2.60 | 0.10.3 | 136 |
| 38.2.61 | 0.10.2 | 137 |
| 38.2.62 | 0.10.1 | 137 |
| 38.2.63 | 0.10 | 137 |
| 38.2.64 | 0.9.17 | 137 |
| 38.2.65 | 0.9.16 | 138 |
| 38.2.66 | 0.9.15 | 138 |
| 38.2.67 | 0.9.14 | 138 |
| 38.2.68 | 0.9.13 | 138 |
| 38.2.69 | 0.9.12 | 138 |
| 38.2.70 | 0.9.11 | 138 |
| 38.2.71 | 0.9.10 | 138 |
| 38.2.72 | 0.9.9 | 139 |
| 38.2.73 | 0.9.8 | 139 |
| 38.2.74 | 0.9.7 | 139 |
| 38.2.75 | 0.9.6 | 139 |
| 38.2.76 | 0.9.5 | 139 |
| 38.2.77 | 0.9.4 | 139 |
| 38.2.78 | 0.9.3 | 139 |
| 38.2.79 | 0.9.2 | 140 |
| 38.2.80 | 0.9.1 | 140 |
| 38.2.81 | 0.9 | 140 |
| 38.2.82 | 0.8.10 | 140 |
| 38.2.83 | 0.8.9 | 140 |
| 38.2.84 | 0.8.8 | 140 |
| 38.2.85 | 0.8.7 | 141 |
| 38.2.86 | 0.8.6 | 141 |
| 38.2.87 | 0.8.5 | 141 |
| 38.2.88 | 0.8.4 | 141 |
| 38.2.89 | 0.8.3 | 141 |
| 38.2.90 | 0.8.2 | 141 |
| 38.2.91 | 0.8.1 | 141 |
| 38.2.92 | 0.8 | 142 |
| 38.2.92.1 | New files | 142 |
| 38.2.92.2 | Existing files | 143 |
| 38.2.92.3 | Additional listings | 143 |
| 38.2.92.3.1 | Listing 0.8.a “tab-form-elemenets” -> “tab-form-elements” | 143 |
| 38.2.92.3.2 | Listing 0.8.b <i>form_view_form_entry_option_class</i> property | 143 |
| 38.2.93 | 0.7.1 | 144 |
| 38.2.94 | 0.7 | 144 |
| 38.2.95 | 0.6.10 | 144 |
| 38.2.96 | 0.6.9 | 144 |
| 38.2.97 | 0.6.8 | 144 |
| 38.2.98 | 0.6.7 | 144 |
| 38.2.99 | 0.6.6 | 145 |
| 38.2.1000 | 0.6.5 | 145 |
| 38.2.1010 | 0.6.4 | 145 |
| 38.2.1020 | 0.6.3 | 145 |
| 38.2.1030 | 0.6.2 | 145 |
| 38.2.1040 | 0.6.1 | 145 |
| 38.2.1050 | 0.6 | 145 |

| | |
|--------------------------|-----|
| 38.2.1060.5.19 | 145 |
| 38.2.1070.5.18 | 146 |
| 38.2.1080.5.17 | 146 |
| 38.2.1090.5.16 | 146 |
| 38.2.1100.5.15 | 146 |
| 38.2.1110.5.14 | 146 |
| 38.2.1120.5.13 | 146 |
| 38.2.1130.5.12 | 146 |
| 38.2.113.1New files | 147 |
| 38.2.113.2Existing files | 147 |
| 38.2.1140.5.11 | 147 |
| 38.2.1150.5.10 | 147 |
| 38.2.1160.5.9 | 147 |
| 38.2.1170.5.8 | 148 |
| 38.2.1180.5.7 | 148 |
| 38.2.1190.5.6 | 148 |
| 38.2.1200.5.5 | 148 |
| 38.2.1210.5.4 | 148 |
| 38.2.1220.5.3 | 148 |
| 38.2.1230.5.2 | 148 |
| 38.2.1240.5.1 | 149 |
| 38.2.1250.5 | 149 |
| 38.2.1260.4.36 | 149 |
| 38.2.1270.4.35 | 149 |
| 38.2.1280.4.34 | 149 |
| 38.2.1290.4.33 | 149 |
| 38.2.1300.4.32 | 150 |
| 38.2.1310.4.31 | 150 |
| 38.2.1320.4.30 | 150 |
| 38.2.1330.4.29 | 150 |
| 38.2.1340.4.28 | 150 |
| 38.2.1350.4.27 | 151 |
| 38.2.1360.4.26 | 151 |
| 38.2.1370.4.25 | 151 |
| 38.2.1380.4.24 | 152 |
| 38.2.1390.4.23 | 152 |
| 38.2.1400.4.22 | 152 |
| 38.2.1410.4.21 | 152 |
| 38.2.1420.4.20 | 152 |
| 38.2.1430.4.19 | 153 |
| 38.2.1440.4.18 | 153 |
| 38.2.1450.4.17 | 153 |
| 38.2.1460.4.16 | 153 |
| 38.2.1470.4.15 | 153 |
| 38.2.1480.4.14 | 153 |
| 38.2.1490.4.13 | 154 |
| 38.2.1500.4.12 | 154 |
| 38.2.1510.4.11 | 154 |
| 38.2.1520.4.10 | 154 |
| 38.2.1530.4.9 | 154 |
| 38.2.1540.4.8 | 154 |
| 38.2.1550.4.7 | 155 |
| 38.2.1560.4.6 | 155 |
| 38.2.1570.4.5 | 155 |

| | |
|---|-----|
| 38.2.1580.4.4 | 155 |
| 38.2.1590.4.3 | 155 |
| 38.2.1600.4.2 | 155 |
| 38.2.1610.4.1 | 156 |
| 38.2.1620.4 | 156 |
| 38.2.1630.3.4 | 157 |
| 38.2.1640.3.3 | 157 |
| 38.2.1650.3.2 | 157 |
| 38.2.1660.3.1 | 157 |
| 38.2.1670.3 | 157 |
| 38.2.1680.2.1 | 158 |
| 38.2.1690.2 | 158 |
| 38.2.1700.1.6 | 158 |
| 38.2.1710.1.5 | 159 |
| 38.2.1720.1.4 | 159 |
| 38.2.1730.1.3 | 159 |
| 38.2.1740.1.2 | 159 |
| 38.2.1750.1.1 | 160 |
| 38.2.1760.1 | 160 |
| 38.3 Licenses | 160 |
| 38.3.1 Bootstrap3 | 160 |
| 38.3.2 Bootstrap 3 DateTime picker | 160 |
| 38.3.3 bootstrap-slider | 160 |
| 38.3.4 django-autoslug | 160 |
| 38.3.5 django-formtools | 160 |
| 38.3.6 django-nine | 161 |
| 38.3.7 django-nonefield | 161 |
| 38.3.8 easy-thumbnails | 161 |
| 38.3.9 Font-awesome | 161 |
| 38.3.10 Foundation 5 | 161 |
| 38.3.11 Foundation 5 DateTime picker | 161 |
| 38.3.12 jQuery Colorbox | 162 |
| 38.3.13 jQuery | 162 |
| 38.3.14 jQueryUI | 162 |
| 38.3.15 jQuery-Slugify-Plugin | 162 |
| 38.3.16 moment.js | 162 |
| 38.3.17 ordereddict | 162 |
| 38.3.18 Pillow | 163 |
| 38.3.19 Requests | 163 |
| 38.3.20 Six | 163 |
| 38.3.21 Unidecode | 163 |
| 38.3.22 vishap | 163 |
| 38.4 fobi package | 164 |
| 38.4.1 Subpackages | 164 |
| 38.4.1.1 fobi.contrib package | 164 |
| 38.4.1.1.1 Subpackages | 164 |
| 38.4.1.1.1.1 fobi.contrib.apps package | 164 |
| 38.4.1.1.1.2 Subpackages | 164 |
| 38.4.1.1.1.3 fobi.contrib.apps.djangocms_integration package | 164 |
| 38.4.1.1.1.4 Submodules | 164 |
| 38.4.1.1.1.5 fobi.contrib.apps.djangocms_integration.apps module | 164 |
| 38.4.1.1.1.6 fobi.contrib.apps.djangocms_integration.cms_plugins module | 164 |
| 38.4.1.1.1.7 fobi.contrib.apps.djangocms_integration.cms_version module | 164 |
| 38.4.1.1.1.8 fobi.contrib.apps.djangocms_integration.conf module | 164 |

| | | |
|---------------|--|-----|
| 38.4.1.1.1.9 | fobi.contrib.apps.djangocms_integration.defaults module | 165 |
| 38.4.1.1.1.10 | fobi.contrib.apps.djangocms_integration.helpers module | 165 |
| 38.4.1.1.1.11 | fobi.contrib.apps.djangocms_integration.models module | 165 |
| 38.4.1.1.1.12 | fobi.contrib.apps.djangocms_integration.settings module | 165 |
| 38.4.1.1.1.13 | Module contents | 165 |
| 38.4.1.1.1.14 | fobi.contrib.apps.drf_integration package | 165 |
| 38.4.1.1.1.15 | Subpackages | 165 |
| 38.4.1.1.1.16 | fobi.contrib.apps.drf_integration.form_elements package | 165 |
| 38.4.1.1.1.17 | Subpackages | 165 |
| 38.4.1.1.1.18 | fobi.contrib.apps.drf_integration.form_elements.content package . | 165 |
| 38.4.1.1.1.19 | Subpackages | 165 |
| 38.4.1.1.1.20 | fobi.contrib.apps.drf_integration.form_elements.content.content_image package | 165 |
| 38.4.1.1.1.21 | Submodules | 165 |
| 38.4.1.1.1.22 | fobi.contrib.apps.drf_integration.form_elements.content.content_image.apps module | 165 |
| 38.4.1.1.1.23 | fobi.contrib.apps.drf_integration.form_elements.content.content_image.base module | 166 |
| 38.4.1.1.1.24 | fobi.contrib.apps.drf_integration.form_elements.content.content_image.fobi_integration_f module | 166 |
| 38.4.1.1.1.25 | Module contents | 166 |
| 38.4.1.1.1.26 | fobi.contrib.apps.drf_integration.form_elements.content.content_image_url package | 166 |
| 38.4.1.1.1.27 | Submodules | 166 |
| 38.4.1.1.1.28 | fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.apps module | 166 |
| 38.4.1.1.1.29 | fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.base module | 167 |
| 38.4.1.1.1.30 | fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.fobi_integrati module | 167 |
| 38.4.1.1.1.31 | Module contents | 167 |
| 38.4.1.1.1.32 | fobi.contrib.apps.drf_integration.form_elements.content.content_markdown package | 167 |
| 38.4.1.1.1.33 | Submodules | 167 |
| 38.4.1.1.1.34 | fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.apps module | 167 |
| 38.4.1.1.1.35 | fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.base module | 168 |
| 38.4.1.1.1.36 | fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.fobi_integrati module | 168 |
| 38.4.1.1.1.37 | Module contents | 168 |
| 38.4.1.1.1.38 | fobi.contrib.apps.drf_integration.form_elements.content.content_richtext package | 168 |
| 38.4.1.1.1.39 | Submodules | 168 |
| 38.4.1.1.1.40 | fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.apps module | 168 |
| 38.4.1.1.1.41 | fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base module | 169 |
| 38.4.1.1.1.42 | fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.fobi_integrati module | 169 |
| 38.4.1.1.1.43 | Module contents | 170 |
| 38.4.1.1.1.44 | fobi.contrib.apps.drf_integration.form_elements.content.content_text package | 170 |
| 38.4.1.1.1.45 | Submodules | 170 |

| | | |
|---------------|---|-----|
| 38.4.1.1.1.46 | fobi.contrib.apps.drf_integration.form_elements.content.content_text.apps module | 170 |
| 38.4.1.1.1.47 | fobi.contrib.apps.drf_integration.form_elements.content.content_text.base module | 170 |
| 38.4.1.1.1.48 | fobi.contrib.apps.drf_integration.form_elements.content.content_text.fobi_integration_for module | 170 |
| 38.4.1.1.1.49 | Module contents | 171 |
| 38.4.1.1.1.50 | fobi.contrib.apps.drf_integration.form_elements.content.content_video package | 171 |
| 38.4.1.1.1.51 | Submodules | 171 |
| 38.4.1.1.1.52 | fobi.contrib.apps.drf_integration.form_elements.content.content_video.apps module | 171 |
| 38.4.1.1.1.53 | fobi.contrib.apps.drf_integration.form_elements.content.content_video.base module | 171 |
| 38.4.1.1.1.54 | fobi.contrib.apps.drf_integration.form_elements.content.content_video.fobi_integration_for module | 171 |
| 38.4.1.1.1.55 | Module contents | 172 |
| 38.4.1.1.1.56 | Module contents | 172 |
| 38.4.1.1.1.57 | fobi.contrib.apps.drf_integration.form_elements.fields package | 172 |
| 38.4.1.1.1.58 | Subpackages | 172 |
| 38.4.1.1.1.59 | fobi.contrib.apps.drf_integration.form_elements.fields.boolean package | 172 |
| 38.4.1.1.1.60 | Submodules | 172 |
| 38.4.1.1.1.61 | fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps module | 172 |
| 38.4.1.1.1.62 | fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base module | 172 |
| 38.4.1.1.1.63 | fobi.contrib.apps.drf_integration.form_elements.fields.boolean.fobi_integration_for_ele module | 172 |
| 38.4.1.1.1.64 | Module contents | 173 |
| 38.4.1.1.1.65 | fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple package | 173 |
| 38.4.1.1.1.66 | Submodules | 173 |
| 38.4.1.1.1.67 | fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.apps module | 173 |
| 38.4.1.1.1.68 | fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.base module | 173 |
| 38.4.1.1.1.69 | fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.fobi_inte module | 174 |
| 38.4.1.1.1.70 | Module contents | 174 |
| 38.4.1.1.1.71 | fobi.contrib.apps.drf_integration.form_elements.fields.date package | 174 |
| 38.4.1.1.1.72 | Submodules | 174 |
| 38.4.1.1.1.73 | fobi.contrib.apps.drf_integration.form_elements.fields.date.apps module | 174 |
| 38.4.1.1.1.74 | fobi.contrib.apps.drf_integration.form_elements.fields.date.base module | 175 |
| 38.4.1.1.1.75 | fobi.contrib.apps.drf_integration.form_elements.fields.date.fobi_integration_form_element module | 175 |
| 38.4.1.1.1.76 | Module contents | 175 |
| 38.4.1.1.1.77 | fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down package | 175 |
| 38.4.1.1.1.78 | Submodules | 175 |
| 38.4.1.1.1.79 | fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps module | 175 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.80 | fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.base module | 176 |
| 38.4.1.1.1.81 | fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.fobi_integration_form_elements module | 176 |
| 38.4.1.1.1.82 | Module contents | 176 |
| 38.4.1.1.1.83 | fobi.contrib.apps.drf_integration.form_elements.fields.datetime package | 176 |
| 38.4.1.1.1.84 | Submodules | 176 |
| 38.4.1.1.1.85 | fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps module | 176 |
| 38.4.1.1.1.86 | fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base module | 177 |
| 38.4.1.1.1.87 | fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_integration_form_elements module | 177 |
| 38.4.1.1.1.88 | Module contents | 177 |
| 38.4.1.1.1.89 | fobi.contrib.apps.drf_integration.form_elements.fields.decimal package | 177 |
| 38.4.1.1.1.90 | Submodules | 177 |
| 38.4.1.1.1.91 | fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps module | 177 |
| 38.4.1.1.1.92 | fobi.contrib.apps.drf_integration.form_elements.fields.decimal.base module | 178 |
| 38.4.1.1.1.93 | fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi_integration_form_elements module | 178 |
| 38.4.1.1.1.94 | Module contents | 178 |
| 38.4.1.1.1.95 | fobi.contrib.apps.drf_integration.form_elements.fields.duration package | 178 |
| 38.4.1.1.1.96 | Submodules | 178 |
| 38.4.1.1.1.97 | fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps module | 178 |
| 38.4.1.1.1.98 | fobi.contrib.apps.drf_integration.form_elements.fields.duration.base module | 179 |
| 38.4.1.1.1.99 | fobi.contrib.apps.drf_integration.form_elements.fields.duration.fobi_integration_form_elements module | 179 |
| 38.4.1.1.1.100 | Module contents | 180 |
| 38.4.1.1.1.101 | fobi.contrib.apps.drf_integration.form_elements.fields.email package | 180 |
| 38.4.1.1.1.102 | Submodules | 180 |
| 38.4.1.1.1.103 | fobi.contrib.apps.drf_integration.form_elements.fields.email.apps module | 180 |
| 38.4.1.1.1.104 | fobi.contrib.apps.drf_integration.form_elements.fields.email.base module | 180 |
| 38.4.1.1.1.105 | fobi.contrib.apps.drf_integration.form_elements.fields.email.fobi_integration_form_elements module | 181 |
| 38.4.1.1.1.106 | Module contents | 181 |
| 38.4.1.1.1.107 | fobi.contrib.apps.drf_integration.form_elements.fields.file package | 181 |
| 38.4.1.1.1.108 | Submodules | 181 |
| 38.4.1.1.1.109 | fobi.contrib.apps.drf_integration.form_elements.fields.file.apps module | 181 |
| 38.4.1.1.1.110 | fobi.contrib.apps.drf_integration.form_elements.fields.file.base module | 181 |
| 38.4.1.1.1.111 | fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_integration_form_elements module | 182 |
| 38.4.1.1.1.112 | Module contents | 183 |

| | | |
|----------------|--|-----|
| 38.4.1.1.1.113 | fobi.contrib.apps.drf_integration.form_elements.fields.float package | 183 |
| 38.4.1.1.1.114 | Submodules | 183 |
| 38.4.1.1.1.115 | fobi.contrib.apps.drf_integration.form_elements.fields.float.apps module | 183 |
| 38.4.1.1.1.116 | fobi.contrib.apps.drf_integration.form_elements.fields.float.base module | 183 |
| 38.4.1.1.1.117 | fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_integration_form_elemen module | 183 |
| 38.4.1.1.1.118 | Module contents | 184 |
| 38.4.1.1.1.119 | fobi.contrib.apps.drf_integration.form_elements.fields.hidden package | 184 |
| 38.4.1.1.1.120 | Submodules | 184 |
| 38.4.1.1.1.121 | fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps module | 184 |
| 38.4.1.1.1.122 | fobi.contrib.apps.drf_integration.form_elements.fields.hidden.base module | 184 |
| 38.4.1.1.1.123 | fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration_form_elemen module | 184 |
| 38.4.1.1.1.124 | Module contents | 185 |
| 38.4.1.1.1.125 | fobi.contrib.apps.drf_integration.form_elements.fields.input package | 185 |
| 38.4.1.1.1.126 | Submodules | 185 |
| 38.4.1.1.1.127 | fobi.contrib.apps.drf_integration.form_elements.fields.input.apps module | 185 |
| 38.4.1.1.1.128 | fobi.contrib.apps.drf_integration.form_elements.fields.input.base module | 185 |
| 38.4.1.1.1.129 | fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration_form_elemen module | 185 |
| 38.4.1.1.1.130 | Module contents | 186 |
| 38.4.1.1.1.131 | fobi.contrib.apps.drf_integration.form_elements.fields.integer package | 186 |
| 38.4.1.1.1.132 | Submodules | 186 |
| 38.4.1.1.1.133 | fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps module | 186 |
| 38.4.1.1.1.134 | fobi.contrib.apps.drf_integration.form_elements.fields.integer.base module | 186 |
| 38.4.1.1.1.135 | fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_integration_form_elemen module | 186 |
| 38.4.1.1.1.136 | Module contents | 187 |
| 38.4.1.1.1.137 | fobi.contrib.apps.drf_integration.form_elements.fields.ip_address package | 187 |
| 38.4.1.1.1.138 | Submodules | 187 |
| 38.4.1.1.1.139 | fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps module | 187 |
| 38.4.1.1.1.140 | fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base module | 187 |
| 38.4.1.1.1.141 | fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration_form_e module | 187 |
| 38.4.1.1.1.142 | Module contents | 188 |
| 38.4.1.1.1.143 | fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean package | 188 |
| 38.4.1.1.1.144 | Submodules | 188 |
| 38.4.1.1.1.145 | fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps module | 188 |

| | | |
|----------------|--|-----|
| 38.4.1.1.1.146 | fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.base module | 188 |
| 38.4.1.1.1.147 | fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form module | 188 |
| 38.4.1.1.1.148 | Module contents | 189 |
| 38.4.1.1.1.149 | fobi.contrib.apps.drf_integration.form_elements.fields.password package | 189 |
| 38.4.1.1.1.150 | Submodules | 189 |
| 38.4.1.1.1.151 | fobi.contrib.apps.drf_integration.form_elements.fields.password.apps module | 189 |
| 38.4.1.1.1.152 | fobi.contrib.apps.drf_integration.form_elements.fields.password.base module | 189 |
| 38.4.1.1.1.153 | fobi.contrib.apps.drf_integration.form_elements.fields.password.fobi_integration_form module | 189 |
| 38.4.1.1.1.154 | Module contents | 190 |
| 38.4.1.1.1.155 | fobi.contrib.apps.drf_integration.form_elements.fields.radio package | 190 |
| 38.4.1.1.1.156 | Submodules | 190 |
| 38.4.1.1.1.157 | fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps module | 190 |
| 38.4.1.1.1.158 | fobi.contrib.apps.drf_integration.form_elements.fields.radio.base module | 190 |
| 38.4.1.1.1.159 | fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_integration_form module | 190 |
| 38.4.1.1.1.160 | Module contents | 191 |
| 38.4.1.1.1.161 | fobi.contrib.apps.drf_integration.form_elements.fields.range_select package | 191 |
| 38.4.1.1.1.162 | Submodules | 191 |
| 38.4.1.1.1.163 | fobi.contrib.apps.drf_integration.form_elements.fields.range_select.apps module | 191 |
| 38.4.1.1.1.164 | fobi.contrib.apps.drf_integration.form_elements.fields.range_select.base module | 191 |
| 38.4.1.1.1.165 | fobi.contrib.apps.drf_integration.form_elements.fields.range_select.fobi_integration_form module | 191 |
| 38.4.1.1.1.166 | Module contents | 192 |
| 38.4.1.1.1.167 | fobi.contrib.apps.drf_integration.form_elements.fields.regex package | 192 |
| 38.4.1.1.1.168 | Submodules | 192 |
| 38.4.1.1.1.169 | fobi.contrib.apps.drf_integration.form_elements.fields.regex.apps module | 192 |
| 38.4.1.1.1.170 | fobi.contrib.apps.drf_integration.form_elements.fields.regex.base module | 192 |
| 38.4.1.1.1.171 | fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_integration_form module | 192 |
| 38.4.1.1.1.172 | Module contents | 193 |
| 38.4.1.1.1.173 | fobi.contrib.apps.drf_integration.form_elements.fields.select package | 193 |
| 38.4.1.1.1.174 | Submodules | 193 |
| 38.4.1.1.1.175 | fobi.contrib.apps.drf_integration.form_elements.fields.select.apps module | 193 |
| 38.4.1.1.1.176 | fobi.contrib.apps.drf_integration.form_elements.fields.select.base module | 193 |
| 38.4.1.1.1.177 | fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi_integration_form module | 193 |
| 38.4.1.1.1.178 | Module contents | 194 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.179 | fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object | |
| | package | 194 |
| 38.4.1.1.1.180 | Submodules | 194 |
| 38.4.1.1.1.181 | fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.apps | |
| | module | 194 |
| 38.4.1.1.1.182 | fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.base | |
| | module | 194 |
| 38.4.1.1.1.183 | fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.fobi_integration_for | |
| | module | 195 |
| 38.4.1.1.1.184 | Module contents | 195 |
| 38.4.1.1.1.185 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple | |
| | package | 195 |
| 38.4.1.1.1.186 | Submodules | 195 |
| 38.4.1.1.1.187 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.apps | |
| | module | 195 |
| 38.4.1.1.1.188 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base | |
| | module | 196 |
| 38.4.1.1.1.189 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.fobi_integration_for | |
| | module | 196 |
| 38.4.1.1.1.190 | Module contents | 197 |
| 38.4.1.1.1.191 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects | |
| | package | 197 |
| 38.4.1.1.1.192 | Submodules | 197 |
| 38.4.1.1.1.193 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.apps | |
| | module | 197 |
| 38.4.1.1.1.194 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.base | |
| | module | 197 |
| 38.4.1.1.1.195 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.fobi | |
| | module | 198 |
| 38.4.1.1.1.196 | Module contents | 199 |
| 38.4.1.1.1.197 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max | |
| | package | 199 |
| 38.4.1.1.1.198 | Submodules | 199 |
| 38.4.1.1.1.199 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.apps | |
| | module | 199 |
| 38.4.1.1.1.200 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.base | |
| | module | 199 |
| 38.4.1.1.1.201 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.fobi_inte | |
| | module | 200 |
| 38.4.1.1.1.202 | Module contents | 200 |
| 38.4.1.1.1.203 | fobi.contrib.apps.drf_integration.form_elements.fields.slider | |
| | package | 200 |
| 38.4.1.1.1.204 | Submodules | 200 |
| 38.4.1.1.1.205 | fobi.contrib.apps.drf_integration.form_elements.fields.slider.apps | |
| | module | 200 |
| 38.4.1.1.1.206 | fobi.contrib.apps.drf_integration.form_elements.fields.slider.base | |
| | module | 201 |
| 38.4.1.1.1.207 | fobi.contrib.apps.drf_integration.form_elements.fields.slider.fobi_integration_form_elemen | |
| | module | 201 |
| 38.4.1.1.1.208 | Module contents | 201 |
| 38.4.1.1.1.209 | fobi.contrib.apps.drf_integration.form_elements.fields.slug package | 201 |
| 38.4.1.1.1.210 | Submodules | 201 |
| 38.4.1.1.1.211 | fobi.contrib.apps.drf_integration.form_elements.fields.slug.apps | |
| | module | 201 |

| | | |
|----------------|--|-----|
| 38.4.1.1.1.212 | fobi.contrib.apps.drf_integration.form_elements.fields.slug.base module | 202 |
| 38.4.1.1.1.213 | fobi.contrib.apps.drf_integration.form_elements.fields.slug.fobi_integration_form_element module | 202 |
| 38.4.1.1.1.214 | Module contents | 202 |
| 38.4.1.1.1.215 | fobi.contrib.apps.drf_integration.form_elements.fields.text package | 202 |
| 38.4.1.1.1.216 | Submodules | 202 |
| 38.4.1.1.1.217 | fobi.contrib.apps.drf_integration.form_elements.fields.text.apps module | 202 |
| 38.4.1.1.1.218 | fobi.contrib.apps.drf_integration.form_elements.fields.text.base module | 203 |
| 38.4.1.1.1.219 | fobi.contrib.apps.drf_integration.form_elements.fields.text.fobi_integration_form_element module | 203 |
| 38.4.1.1.1.220 | Module contents | 203 |
| 38.4.1.1.1.221 | fobi.contrib.apps.drf_integration.form_elements.fields.textarea package | 203 |
| 38.4.1.1.1.222 | Submodules | 203 |
| 38.4.1.1.1.223 | fobi.contrib.apps.drf_integration.form_elements.fields.textarea.apps module | 203 |
| 38.4.1.1.1.224 | fobi.contrib.apps.drf_integration.form_elements.fields.textarea.base module | 204 |
| 38.4.1.1.1.225 | fobi.contrib.apps.drf_integration.form_elements.fields.textarea.fobi_integration_form_element module | 204 |
| 38.4.1.1.1.226 | Module contents | 204 |
| 38.4.1.1.1.227 | fobi.contrib.apps.drf_integration.form_elements.fields.time package | 204 |
| 38.4.1.1.1.228 | Submodules | 204 |
| 38.4.1.1.1.229 | fobi.contrib.apps.drf_integration.form_elements.fields.time.apps module | 204 |
| 38.4.1.1.1.230 | fobi.contrib.apps.drf_integration.form_elements.fields.time.base module | 205 |
| 38.4.1.1.1.231 | fobi.contrib.apps.drf_integration.form_elements.fields.time.fobi_integration_form_element module | 205 |
| 38.4.1.1.1.232 | Module contents | 205 |
| 38.4.1.1.1.233 | fobi.contrib.apps.drf_integration.form_elements.fields.url package | 205 |
| 38.4.1.1.1.234 | Submodules | 205 |
| 38.4.1.1.1.235 | fobi.contrib.apps.drf_integration.form_elements.fields.url.apps module | 205 |
| 38.4.1.1.1.236 | fobi.contrib.apps.drf_integration.form_elements.fields.url.base module | 206 |
| 38.4.1.1.1.237 | fobi.contrib.apps.drf_integration.form_elements.fields.url.fobi_integration_form_elements module | 206 |
| 38.4.1.1.1.238 | Module contents | 207 |
| 38.4.1.1.1.239 | Module contents | 207 |
| 38.4.1.1.1.240 | Module contents | 207 |
| 38.4.1.1.1.241 | fobi.contrib.apps.drf_integration.form_handlers package | 207 |
| 38.4.1.1.1.242 | Subpackages | 207 |
| 38.4.1.1.1.243 | fobi.contrib.apps.drf_integration.form_handlers.db_store package | 207 |
| 38.4.1.1.1.244 | Submodules | 207 |
| 38.4.1.1.1.245 | fobi.contrib.apps.drf_integration.form_handlers.db_store.apps module | 207 |
| 38.4.1.1.1.246 | fobi.contrib.apps.drf_integration.form_handlers.db_store.base module | 207 |
| 38.4.1.1.1.247 | fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration_form_handlers module | 207 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.248 | Module contents | 208 |
| 38.4.1.1.1.249 | fobi.contrib.apps.drf_integration.form_handlers.http_repost package | 208 |
| 38.4.1.1.1.250 | Submodules | 208 |
| 38.4.1.1.1.251 | fobi.contrib.apps.drf_integration.form_handlers.http_repost.apps module | 208 |
| 38.4.1.1.1.252 | fobi.contrib.apps.drf_integration.form_handlers.http_repost.base module | 208 |
| 38.4.1.1.1.253 | fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handlers module | 208 |
| 38.4.1.1.1.254 | Module contents | 209 |
| 38.4.1.1.1.255 | fobi.contrib.apps.drf_integration.form_handlers.mail package . . . | 209 |
| 38.4.1.1.1.256 | Submodules | 209 |
| 38.4.1.1.1.257 | fobi.contrib.apps.drf_integration.form_handlers.mail.apps module | 209 |
| 38.4.1.1.1.258 | fobi.contrib.apps.drf_integration.form_handlers.mail.base module | 209 |
| 38.4.1.1.1.259 | fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_integration_form_handlers module | 209 |
| 38.4.1.1.1.260 | Module contents | 210 |
| 38.4.1.1.1.261 | Module contents | 210 |
| 38.4.1.1.1.262 | Submodules | 210 |
| 38.4.1.1.1.263 | fobi.contrib.apps.drf_integration.apps module | 210 |
| 38.4.1.1.1.264 | fobi.contrib.apps.drf_integration.base module | 210 |
| 38.4.1.1.1.265 | fobi.contrib.apps.drf_integration.dynamic module | 212 |
| 38.4.1.1.1.266 | fobi.contrib.apps.drf_integration.fields module | 212 |
| 38.4.1.1.1.267 | fobi.contrib.apps.drf_integration.helpers module | 214 |
| 38.4.1.1.1.268 | fobi.contrib.apps.drf_integration.metadata module | 214 |
| 38.4.1.1.1.269 | fobi.contrib.apps.drf_integration.serializers module | 214 |
| 38.4.1.1.1.270 | fobi.contrib.apps.drf_integration.urls module | 215 |
| 38.4.1.1.1.271 | fobi.contrib.apps.drf_integration.utils module | 215 |
| 38.4.1.1.1.272 | fobi.contrib.apps.drf_integration.views module | 215 |
| 38.4.1.1.1.273 | Module contents | 216 |
| 38.4.1.1.1.274 | fobi.contrib.apps.feincms_integration package | 216 |
| 38.4.1.1.1.275 | Submodules | 216 |
| 38.4.1.1.1.276 | fobi.contrib.apps.feincms_integration.apps module | 216 |
| 38.4.1.1.1.277 | fobi.contrib.apps.feincms_integration.conf module | 216 |
| 38.4.1.1.1.278 | fobi.contrib.apps.feincms_integration.defaults module | 216 |
| 38.4.1.1.1.279 | fobi.contrib.apps.feincms_integration.helpers module | 216 |
| 38.4.1.1.1.280 | fobi.contrib.apps.feincms_integration.settings module | 216 |
| 38.4.1.1.1.281 | fobi.contrib.apps.feincms_integration.widgets module | 217 |
| 38.4.1.1.1.282 | Module contents | 218 |
| 38.4.1.1.1.283 | fobi.contrib.apps.mezzanine_integration package | 218 |
| 38.4.1.1.1.284 | Submodules | 218 |
| 38.4.1.1.1.285 | fobi.contrib.apps.mezzanine_integration.admin module | 218 |
| 38.4.1.1.1.286 | fobi.contrib.apps.mezzanine_integration.apps module | 218 |
| 38.4.1.1.1.287 | fobi.contrib.apps.mezzanine_integration.conf module | 218 |
| 38.4.1.1.1.288 | fobi.contrib.apps.mezzanine_integration.defaults module | 219 |
| 38.4.1.1.1.289 | fobi.contrib.apps.mezzanine_integration.helpers module | 219 |
| 38.4.1.1.1.290 | fobi.contrib.apps.mezzanine_integration.models module | 219 |
| 38.4.1.1.1.291 | fobi.contrib.apps.mezzanine_integration.page_processors module . | 219 |
| 38.4.1.1.1.292 | fobi.contrib.apps.mezzanine_integration.settings module | 219 |
| 38.4.1.1.1.293 | Module contents | 220 |
| 38.4.1.1.1.294 | Module contents | 220 |
| 38.4.1.1.1.295 | fobi.contrib.plugins package | 220 |
| 38.4.1.1.1.296 | Subpackages | 220 |
| 38.4.1.1.1.297 | fobi.contrib.plugins.form_elements package | 220 |

| | |
|--|-----|
| 38.4.1.1.1.298 Subpackages | 220 |
| 38.4.1.1.1.299 fobi.contrib.plugins.form_elements.content package | 220 |
| 38.4.1.1.1.300 Subpackages | 220 |
| 38.4.1.1.1.301 fobi.contrib.plugins.form_elements.content.content_image package | 220 |
| 38.4.1.1.1.302 Submodules | 220 |
| 38.4.1.1.1.303 fobi.contrib.plugins.form_elements.content.content_image.apps module | 220 |
| 38.4.1.1.1.304 fobi.contrib.plugins.form_elements.content.content_image.base module | 220 |
| 38.4.1.1.1.305 fobi.contrib.plugins.form_elements.content.content_image.conf module | 221 |
| 38.4.1.1.1.306 fobi.contrib.plugins.form_elements.content.content_image.defaults module | 221 |
| 38.4.1.1.1.307 fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements module | 221 |
| 38.4.1.1.1.308 fobi.contrib.plugins.form_elements.content.content_image.forms module | 222 |
| 38.4.1.1.1.309 fobi.contrib.plugins.form_elements.content.content_image.helpers module | 223 |
| 38.4.1.1.1.310 fobi.contrib.plugins.form_elements.content.content_image.settings module | 223 |
| 38.4.1.1.1.311 Module contents | 223 |
| 38.4.1.1.1.312 fobi.contrib.plugins.form_elements.content.content_image_url package | 223 |
| 38.4.1.1.1.313 Submodules | 223 |
| 38.4.1.1.1.314 fobi.contrib.plugins.form_elements.content.content_image_url.apps module | 223 |
| 38.4.1.1.1.315 fobi.contrib.plugins.form_elements.content.content_image_url.base module | 223 |
| 38.4.1.1.1.316 fobi.contrib.plugins.form_elements.content.content_image_url.conf module | 224 |
| 38.4.1.1.1.317 fobi.contrib.plugins.form_elements.content.content_image_url.defaults module | 224 |
| 38.4.1.1.1.318 fobi.contrib.plugins.form_elements.content.content_image_url.fobi_form_elements module | 224 |
| 38.4.1.1.1.319 fobi.contrib.plugins.form_elements.content.content_image_url.forms module | 225 |
| 38.4.1.1.1.320 fobi.contrib.plugins.form_elements.content.content_image_url.settings module | 225 |
| 38.4.1.1.1.321 Module contents | 226 |
| 38.4.1.1.1.322 fobi.contrib.plugins.form_elements.content.content_markdown package | 226 |
| 38.4.1.1.1.323 Submodules | 226 |
| 38.4.1.1.1.324 fobi.contrib.plugins.form_elements.content.content_markdown.apps module | 226 |
| 38.4.1.1.1.325 fobi.contrib.plugins.form_elements.content.content_markdown.base module | 226 |
| 38.4.1.1.1.326 fobi.contrib.plugins.form_elements.content.content_markdown.fobi_form_elements module | 227 |
| 38.4.1.1.1.327 fobi.contrib.plugins.form_elements.content.content_markdown.forms module | 228 |
| 38.4.1.1.1.328 fobi.contrib.plugins.form_elements.content.content_markdown.widgets module | 228 |
| 38.4.1.1.1.329 Module contents | 228 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.330 | fobi.contrib.plugins.form_elements.content.content_richtext package | 228 |
| 38.4.1.1.1.331 | Submodules | 228 |
| 38.4.1.1.1.332 | fobi.contrib.plugins.form_elements.content.content_richtext.apps module | 228 |
| 38.4.1.1.1.333 | fobi.contrib.plugins.form_elements.content.content_richtext.base module | 229 |
| 38.4.1.1.1.334 | fobi.contrib.plugins.form_elements.content.content_richtext.fobi_form_elements module | 229 |
| 38.4.1.1.1.335 | fobi.contrib.plugins.form_elements.content.content_richtext.forms module | 231 |
| 38.4.1.1.1.336 | fobi.contrib.plugins.form_elements.content.content_richtext.widgets module | 231 |
| 38.4.1.1.1.337 | Module contents | 231 |
| 38.4.1.1.1.338 | fobi.contrib.plugins.form_elements.content.content_text package | 231 |
| 38.4.1.1.1.339 | Submodules | 231 |
| 38.4.1.1.1.340 | fobi.contrib.plugins.form_elements.content.content_text.apps module | 231 |
| 38.4.1.1.1.341 | fobi.contrib.plugins.form_elements.content.content_text.base module | 232 |
| 38.4.1.1.1.342 | fobi.contrib.plugins.form_elements.content.content_text.conf module | 232 |
| 38.4.1.1.1.343 | fobi.contrib.plugins.form_elements.content.content_text.defaults module | 233 |
| 38.4.1.1.1.344 | fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements module | 233 |
| 38.4.1.1.1.345 | fobi.contrib.plugins.form_elements.content.content_text.forms module | 233 |
| 38.4.1.1.1.346 | fobi.contrib.plugins.form_elements.content.content_text.settings module | 234 |
| 38.4.1.1.1.347 | Module contents | 234 |
| 38.4.1.1.1.348 | fobi.contrib.plugins.form_elements.content.content_video package | 234 |
| 38.4.1.1.1.349 | Submodules | 234 |
| 38.4.1.1.1.350 | fobi.contrib.plugins.form_elements.content.content_video.apps module | 234 |
| 38.4.1.1.1.351 | fobi.contrib.plugins.form_elements.content.content_video.base module | 234 |
| 38.4.1.1.1.352 | fobi.contrib.plugins.form_elements.content.content_video.conf module | 235 |
| 38.4.1.1.1.353 | fobi.contrib.plugins.form_elements.content.content_video.defaults module | 235 |
| 38.4.1.1.1.354 | fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements module | 235 |
| 38.4.1.1.1.355 | fobi.contrib.plugins.form_elements.content.content_video.forms module | 236 |
| 38.4.1.1.1.356 | fobi.contrib.plugins.form_elements.content.content_video.settings module | 237 |
| 38.4.1.1.1.357 | Module contents | 237 |
| 38.4.1.1.1.358 | Module contents | 237 |
| 38.4.1.1.1.359 | fobi.contrib.plugins.form_elements.fields package | 237 |
| 38.4.1.1.1.360 | Subpackages | 237 |
| 38.4.1.1.1.361 | fobi.contrib.plugins.form_elements.fields.boolean package | 237 |
| 38.4.1.1.1.362 | Submodules | 237 |
| 38.4.1.1.1.363 | fobi.contrib.plugins.form_elements.fields.boolean.apps module | 237 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.364 | fobi.contrib.plugins.form_elements.fields.boolean.base module . . . | 237 |
| 38.4.1.1.1.365 | fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements module | 237 |
| 38.4.1.1.1.366 | fobi.contrib.plugins.form_elements.fields.boolean.forms module . . | 238 |
| 38.4.1.1.1.367 | Module contents | 238 |
| 38.4.1.1.1.368 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple package | 238 |
| 38.4.1.1.1.369 | Submodules | 238 |
| 38.4.1.1.1.370 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps module | 238 |
| 38.4.1.1.1.371 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.base module | 239 |
| 38.4.1.1.1.372 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf module | 239 |
| 38.4.1.1.1.373 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.defaults module | 240 |
| 38.4.1.1.1.374 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements module | 240 |
| 38.4.1.1.1.375 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms module | 241 |
| 38.4.1.1.1.376 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.settings module | 241 |
| 38.4.1.1.1.377 | Module contents | 241 |
| 38.4.1.1.1.378 | fobi.contrib.plugins.form_elements.fields.date package | 241 |
| 38.4.1.1.1.379 | Submodules | 241 |
| 38.4.1.1.1.380 | fobi.contrib.plugins.form_elements.fields.date.apps module | 241 |
| 38.4.1.1.1.381 | fobi.contrib.plugins.form_elements.fields.date.base module | 242 |
| 38.4.1.1.1.382 | fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements module | 242 |
| 38.4.1.1.1.383 | fobi.contrib.plugins.form_elements.fields.date.forms module . . . | 243 |
| 38.4.1.1.1.384 | fobi.contrib.plugins.form_elements.fields.date.widgets module . . | 243 |
| 38.4.1.1.1.385 | Module contents | 243 |
| 38.4.1.1.1.386 | fobi.contrib.plugins.form_elements.fields.date_drop_down package | 243 |
| 38.4.1.1.1.387 | Submodules | 243 |
| 38.4.1.1.1.388 | fobi.contrib.plugins.form_elements.fields.date_drop_down.apps module | 243 |
| 38.4.1.1.1.389 | fobi.contrib.plugins.form_elements.fields.date_drop_down.base module | 244 |
| 38.4.1.1.1.390 | fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements module | 244 |
| 38.4.1.1.1.391 | fobi.contrib.plugins.form_elements.fields.date_drop_down.forms module | 245 |
| 38.4.1.1.1.392 | Module contents | 245 |
| 38.4.1.1.1.393 | fobi.contrib.plugins.form_elements.fields.datetime package | 245 |
| 38.4.1.1.1.394 | Submodules | 245 |
| 38.4.1.1.1.395 | fobi.contrib.plugins.form_elements.fields.datetime.apps module . . | 245 |
| 38.4.1.1.1.396 | fobi.contrib.plugins.form_elements.fields.datetime.base module . . | 246 |
| 38.4.1.1.1.397 | fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements module | 246 |
| 38.4.1.1.1.398 | fobi.contrib.plugins.form_elements.fields.datetime.forms module . | 247 |
| 38.4.1.1.1.399 | fobi.contrib.plugins.form_elements.fields.datetime.widgets module | 247 |
| 38.4.1.1.1.400 | Module contents | 247 |
| 38.4.1.1.1.401 | fobi.contrib.plugins.form_elements.fields.decimal package | 247 |
| 38.4.1.1.1.402 | Submodules | 247 |

| | |
|--|-----|
| 38.4.1.1.1.403 fobi.contrib.plugins.form_elements.fields.decimal.apps module . . | 247 |
| 38.4.1.1.1.404 fobi.contrib.plugins.form_elements.fields.decimal.base module . . | 248 |
| 38.4.1.1.1.405 fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements module | 248 |
| 38.4.1.1.1.406 fobi.contrib.plugins.form_elements.fields.decimal.forms module . | 249 |
| 38.4.1.1.1.407 Module contents | 249 |
| 38.4.1.1.1.408 fobi.contrib.plugins.form_elements.fields.duration package | 249 |
| 38.4.1.1.1.409 Submodules | 249 |
| 38.4.1.1.1.410 fobi.contrib.plugins.form_elements.fields.duration.apps module . . | 249 |
| 38.4.1.1.1.411 fobi.contrib.plugins.form_elements.fields.duration.base module . . | 250 |
| 38.4.1.1.1.412 fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements module | 250 |
| 38.4.1.1.1.413 fobi.contrib.plugins.form_elements.fields.duration.forms module . | 251 |
| 38.4.1.1.1.414 Module contents | 251 |
| 38.4.1.1.1.415 fobi.contrib.plugins.form_elements.fields.email package | 251 |
| 38.4.1.1.1.416 Submodules | 251 |
| 38.4.1.1.1.417 fobi.contrib.plugins.form_elements.fields.email.apps module . . . | 251 |
| 38.4.1.1.1.418 fobi.contrib.plugins.form_elements.fields.email.base module . . . | 252 |
| 38.4.1.1.1.419 fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements module | 252 |
| 38.4.1.1.1.420 fobi.contrib.plugins.form_elements.fields.email.forms module . . | 253 |
| 38.4.1.1.1.421 Module contents | 253 |
| 38.4.1.1.1.422 fobi.contrib.plugins.form_elements.fields.file package | 253 |
| 38.4.1.1.1.423 Submodules | 253 |
| 38.4.1.1.1.424 fobi.contrib.plugins.form_elements.fields.file.apps module | 253 |
| 38.4.1.1.1.425 fobi.contrib.plugins.form_elements.fields.file.base module | 253 |
| 38.4.1.1.1.426 fobi.contrib.plugins.form_elements.fields.file.conf module | 254 |
| 38.4.1.1.1.427 fobi.contrib.plugins.form_elements.fields.file.defaults module . . . | 255 |
| 38.4.1.1.1.428 fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements module | 255 |
| 38.4.1.1.1.429 fobi.contrib.plugins.form_elements.fields.file.forms module | 256 |
| 38.4.1.1.1.430 fobi.contrib.plugins.form_elements.fields.file.settings module . . . | 256 |
| 38.4.1.1.1.431 Module contents | 256 |
| 38.4.1.1.1.432 fobi.contrib.plugins.form_elements.fields.float package | 256 |
| 38.4.1.1.1.433 Submodules | 256 |
| 38.4.1.1.1.434 fobi.contrib.plugins.form_elements.fields.float.apps module | 256 |
| 38.4.1.1.1.435 fobi.contrib.plugins.form_elements.fields.float.base module | 257 |
| 38.4.1.1.1.436 fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements module | 257 |
| 38.4.1.1.1.437 fobi.contrib.plugins.form_elements.fields.float.forms module . . . | 258 |
| 38.4.1.1.1.438 Module contents | 258 |
| 38.4.1.1.1.439 fobi.contrib.plugins.form_elements.fields.hidden package | 258 |
| 38.4.1.1.1.440 Submodules | 258 |
| 38.4.1.1.1.441 fobi.contrib.plugins.form_elements.fields.hidden.apps module . . . | 258 |
| 38.4.1.1.1.442 fobi.contrib.plugins.form_elements.fields.hidden.base module . . . | 258 |
| 38.4.1.1.1.443 fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements module | 259 |
| 38.4.1.1.1.444 fobi.contrib.plugins.form_elements.fields.hidden.forms module . . | 259 |
| 38.4.1.1.1.445 Module contents | 260 |
| 38.4.1.1.1.446 fobi.contrib.plugins.form_elements.fields.input package | 260 |
| 38.4.1.1.1.447 Submodules | 260 |
| 38.4.1.1.1.448 fobi.contrib.plugins.form_elements.fields.input.apps module . . . | 260 |
| 38.4.1.1.1.449 fobi.contrib.plugins.form_elements.fields.input.base module . . . | 260 |
| 38.4.1.1.1.450 fobi.contrib.plugins.form_elements.fields.input.constants module . | 260 |

| | |
|---|-----|
| 38.4.1.1.1.451 fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements module | 260 |
| 38.4.1.1.1.452 fobi.contrib.plugins.form_elements.fields.input.forms module . . . | 261 |
| 38.4.1.1.1.453 Module contents | 262 |
| 38.4.1.1.1.454 fobi.contrib.plugins.form_elements.fields.integer package | 262 |
| 38.4.1.1.1.455 Submodules | 262 |
| 38.4.1.1.1.456 fobi.contrib.plugins.form_elements.fields.integer.apps module . . | 262 |
| 38.4.1.1.1.457 fobi.contrib.plugins.form_elements.fields.integer.base module . . | 262 |
| 38.4.1.1.1.458 fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements module | 262 |
| 38.4.1.1.1.459 fobi.contrib.plugins.form_elements.fields.integer.forms module . . | 263 |
| 38.4.1.1.1.460 Module contents | 263 |
| 38.4.1.1.1.461 fobi.contrib.plugins.form_elements.fields.ip_address package . . . | 263 |
| 38.4.1.1.1.462 Submodules | 263 |
| 38.4.1.1.1.463 fobi.contrib.plugins.form_elements.fields.ip_address.apps module | 263 |
| 38.4.1.1.1.464 fobi.contrib.plugins.form_elements.fields.ip_address.base module | 263 |
| 38.4.1.1.1.465 fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements module | 264 |
| 38.4.1.1.1.466 fobi.contrib.plugins.form_elements.fields.ip_address.forms module | 264 |
| 38.4.1.1.1.467 Module contents | 265 |
| 38.4.1.1.1.468 fobi.contrib.plugins.form_elements.fields.null_boolean package . | 265 |
| 38.4.1.1.1.469 Submodules | 265 |
| 38.4.1.1.1.470 fobi.contrib.plugins.form_elements.fields.null_boolean.apps module | 265 |
| 38.4.1.1.1.471 fobi.contrib.plugins.form_elements.fields.null_boolean.base module | 265 |
| 38.4.1.1.1.472 fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements module | 265 |
| 38.4.1.1.1.473 fobi.contrib.plugins.form_elements.fields.null_boolean.forms module | 266 |
| 38.4.1.1.1.474 Module contents | 266 |
| 38.4.1.1.1.475 fobi.contrib.plugins.form_elements.fields.password package . . . | 266 |
| 38.4.1.1.1.476 Submodules | 266 |
| 38.4.1.1.1.477 fobi.contrib.plugins.form_elements.fields.password.apps module . | 266 |
| 38.4.1.1.1.478 fobi.contrib.plugins.form_elements.fields.password.base module . | 267 |
| 38.4.1.1.1.479 fobi.contrib.plugins.form_elements.fields.password.fobi_form_elements module | 267 |
| 38.4.1.1.1.480 fobi.contrib.plugins.form_elements.fields.password.forms module | 268 |
| 38.4.1.1.1.481 Module contents | 268 |
| 38.4.1.1.1.482 fobi.contrib.plugins.form_elements.fields.radio package | 268 |
| 38.4.1.1.1.483 Submodules | 268 |
| 38.4.1.1.1.484 fobi.contrib.plugins.form_elements.fields.radio.apps module . . . | 268 |
| 38.4.1.1.1.485 fobi.contrib.plugins.form_elements.fields.radio.base module . . . | 268 |
| 38.4.1.1.1.486 fobi.contrib.plugins.form_elements.fields.radio.conf module . . . | 269 |
| 38.4.1.1.1.487 fobi.contrib.plugins.form_elements.fields.radio.defaults module . | 269 |
| 38.4.1.1.1.488 fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements module | 269 |
| 38.4.1.1.1.489 fobi.contrib.plugins.form_elements.fields.radio.forms module . . . | 270 |
| 38.4.1.1.1.490 fobi.contrib.plugins.form_elements.fields.radio.settings module . . | 271 |
| 38.4.1.1.1.491 Module contents | 271 |
| 38.4.1.1.1.492 fobi.contrib.plugins.form_elements.fields.range_select package . . | 271 |
| 38.4.1.1.1.493 Submodules | 271 |
| 38.4.1.1.1.494 fobi.contrib.plugins.form_elements.fields.range_select.apps module | 271 |
| 38.4.1.1.1.495 fobi.contrib.plugins.form_elements.fields.range_select.base module | 271 |
| 38.4.1.1.1.496 fobi.contrib.plugins.form_elements.fields.range_select.conf module | 271 |

| | | |
|----------------|--|-----|
| 38.4.1.1.1.497 | fobi.contrib.plugins.form_elements.fields.range_select.defaults | |
| | module | 272 |
| 38.4.1.1.1.498 | fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements | |
| | module | 272 |
| 38.4.1.1.1.499 | fobi.contrib.plugins.form_elements.fields.range_select.forms | |
| | module | 273 |
| 38.4.1.1.1.500 | fobi.contrib.plugins.form_elements.fields.range_select.settings | |
| | module | 273 |
| 38.4.1.1.1.501 | Module contents | 273 |
| 38.4.1.1.1.502 | fobi.contrib.plugins.form_elements.fields.regex package | 273 |
| 38.4.1.1.1.503 | Submodules | 273 |
| 38.4.1.1.1.504 | fobi.contrib.plugins.form_elements.fields.regex.apps module | 273 |
| 38.4.1.1.1.505 | fobi.contrib.plugins.form_elements.fields.regex.base module | 274 |
| 38.4.1.1.1.506 | fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements | |
| | module | 274 |
| 38.4.1.1.1.507 | fobi.contrib.plugins.form_elements.fields.regex.forms module | 275 |
| 38.4.1.1.1.508 | Module contents | 275 |
| 38.4.1.1.1.509 | fobi.contrib.plugins.form_elements.fields.select package | 275 |
| 38.4.1.1.1.510 | Submodules | 275 |
| 38.4.1.1.1.511 | fobi.contrib.plugins.form_elements.fields.select.apps module | 275 |
| 38.4.1.1.1.512 | fobi.contrib.plugins.form_elements.fields.select.base module | 275 |
| 38.4.1.1.1.513 | fobi.contrib.plugins.form_elements.fields.select.conf module | 276 |
| 38.4.1.1.1.514 | fobi.contrib.plugins.form_elements.fields.select.defaults module | 276 |
| 38.4.1.1.1.515 | fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements | |
| | module | 276 |
| 38.4.1.1.1.516 | fobi.contrib.plugins.form_elements.fields.select.forms module | 277 |
| 38.4.1.1.1.517 | fobi.contrib.plugins.form_elements.fields.select.settings module | 278 |
| 38.4.1.1.1.518 | Module contents | 278 |
| 38.4.1.1.1.519 | fobi.contrib.plugins.form_elements.fields.select_model_object | |
| | package | 278 |
| 38.4.1.1.1.520 | Submodules | 278 |
| 38.4.1.1.1.521 | fobi.contrib.plugins.form_elements.fields.select_model_object.apps | |
| | module | 278 |
| 38.4.1.1.1.522 | fobi.contrib.plugins.form_elements.fields.select_model_object.base | |
| | module | 278 |
| 38.4.1.1.1.523 | fobi.contrib.plugins.form_elements.fields.select_model_object.conf | |
| | module | 279 |
| 38.4.1.1.1.524 | fobi.contrib.plugins.form_elements.fields.select_model_object.defaults | |
| | module | 279 |
| 38.4.1.1.1.525 | fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements | |
| | module | 279 |
| 38.4.1.1.1.526 | fobi.contrib.plugins.form_elements.fields.select_model_object.forms | |
| | module | 280 |
| 38.4.1.1.1.527 | fobi.contrib.plugins.form_elements.fields.select_model_object.settings | |
| | module | 280 |
| 38.4.1.1.1.528 | Module contents | 280 |
| 38.4.1.1.1.529 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object | |
| | package | 280 |
| 38.4.1.1.1.530 | Submodules | 280 |
| 38.4.1.1.1.531 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps | |
| | module | 280 |
| 38.4.1.1.1.532 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base | |
| | module | 280 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.533 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf module | 281 |
| 38.4.1.1.1.534 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults module | 281 |
| 38.4.1.1.1.535 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.fobi_form_elements module | 281 |
| 38.4.1.1.1.536 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms module | 282 |
| 38.4.1.1.1.537 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.settings module | 282 |
| 38.4.1.1.1.538 | Module contents | 282 |
| 38.4.1.1.1.539 | fobi.contrib.plugins.form_elements.fields.select_multiple package | 282 |
| 38.4.1.1.1.540 | Submodules | 282 |
| 38.4.1.1.1.541 | fobi.contrib.plugins.form_elements.fields.select_multiple.apps module | 282 |
| 38.4.1.1.1.542 | fobi.contrib.plugins.form_elements.fields.select_multiple.base module | 282 |
| 38.4.1.1.1.543 | fobi.contrib.plugins.form_elements.fields.select_multiple.conf module | 283 |
| 38.4.1.1.1.544 | fobi.contrib.plugins.form_elements.fields.select_multiple.defaults module | 284 |
| 38.4.1.1.1.545 | fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements module | 284 |
| 38.4.1.1.1.546 | fobi.contrib.plugins.form_elements.fields.select_multiple.forms module | 285 |
| 38.4.1.1.1.547 | fobi.contrib.plugins.form_elements.fields.select_multiple.settings module | 285 |
| 38.4.1.1.1.548 | Module contents | 285 |
| 38.4.1.1.1.549 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects package | 285 |
| 38.4.1.1.1.550 | Submodules | 285 |
| 38.4.1.1.1.551 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps module | 285 |
| 38.4.1.1.1.552 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.base module | 286 |
| 38.4.1.1.1.553 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf module | 286 |
| 38.4.1.1.1.554 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults module | 287 |
| 38.4.1.1.1.555 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_eleme module | 287 |
| 38.4.1.1.1.556 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms module | 287 |
| 38.4.1.1.1.557 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.settings module | 288 |
| 38.4.1.1.1.558 | Module contents | 288 |
| 38.4.1.1.1.559 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects package | 288 |
| 38.4.1.1.1.560 | Submodules | 288 |
| 38.4.1.1.1.561 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps module | 288 |
| 38.4.1.1.1.562 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.base module | 288 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.563 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf module | 289 |
| 38.4.1.1.1.564 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults module | 289 |
| 38.4.1.1.1.565 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form module | 289 |
| 38.4.1.1.1.566 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms module | 290 |
| 38.4.1.1.1.567 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.settings module | 290 |
| 38.4.1.1.1.568 | Module contents | 290 |
| 38.4.1.1.1.569 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max package | 290 |
| 38.4.1.1.1.570 | Submodules | 290 |
| 38.4.1.1.1.571 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.apps module | 290 |
| 38.4.1.1.1.572 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base module | 290 |
| 38.4.1.1.1.573 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.conf module | 291 |
| 38.4.1.1.1.574 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.defaults module | 292 |
| 38.4.1.1.1.575 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fields module | 292 |
| 38.4.1.1.1.576 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements module | 292 |
| 38.4.1.1.1.577 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms module | 293 |
| 38.4.1.1.1.578 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.settings module | 293 |
| 38.4.1.1.1.579 | Module contents | 293 |
| 38.4.1.1.1.580 | fobi.contrib.plugins.form_elements.fields.slider package | 293 |
| 38.4.1.1.1.581 | Submodules | 293 |
| 38.4.1.1.1.582 | fobi.contrib.plugins.form_elements.fields.slider.apps module | 293 |
| 38.4.1.1.1.583 | fobi.contrib.plugins.form_elements.fields.slider.base module | 294 |
| 38.4.1.1.1.584 | fobi.contrib.plugins.form_elements.fields.slider.conf module | 294 |
| 38.4.1.1.1.585 | fobi.contrib.plugins.form_elements.fields.slider.constants module | 295 |
| 38.4.1.1.1.586 | fobi.contrib.plugins.form_elements.fields.slider.defaults module | 295 |
| 38.4.1.1.1.587 | fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements module | 295 |
| 38.4.1.1.1.588 | fobi.contrib.plugins.form_elements.fields.slider.forms module | 295 |
| 38.4.1.1.1.589 | fobi.contrib.plugins.form_elements.fields.slider.helpers module | 296 |
| 38.4.1.1.1.590 | fobi.contrib.plugins.form_elements.fields.slider.settings module | 296 |
| 38.4.1.1.1.591 | fobi.contrib.plugins.form_elements.fields.slider.widgets module | 296 |
| 38.4.1.1.1.592 | Module contents | 296 |
| 38.4.1.1.1.593 | fobi.contrib.plugins.form_elements.fields.slug package | 296 |
| 38.4.1.1.1.594 | Submodules | 296 |
| 38.4.1.1.1.595 | fobi.contrib.plugins.form_elements.fields.slug.apps module | 296 |
| 38.4.1.1.1.596 | fobi.contrib.plugins.form_elements.fields.slug.base module | 297 |
| 38.4.1.1.1.597 | fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements module | 297 |
| 38.4.1.1.1.598 | fobi.contrib.plugins.form_elements.fields.slug.forms module | 298 |
| 38.4.1.1.1.599 | Module contents | 298 |
| 38.4.1.1.1.600 | fobi.contrib.plugins.form_elements.fields.text package | 298 |

| | |
|--|-----|
| 38.4.1.1.1.601 Submodules | 298 |
| 38.4.1.1.1.602 fobi.contrib.plugins.form_elements.fields.text.apps module | 298 |
| 38.4.1.1.1.603 fobi.contrib.plugins.form_elements.fields.text.base module | 298 |
| 38.4.1.1.1.604 fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements module | 299 |
| 38.4.1.1.1.605 fobi.contrib.plugins.form_elements.fields.text.forms module . . . | 299 |
| 38.4.1.1.1.606 Module contents | 300 |
| 38.4.1.1.1.607 fobi.contrib.plugins.form_elements.fields.textarea package | 300 |
| 38.4.1.1.1.608 Submodules | 300 |
| 38.4.1.1.1.609 fobi.contrib.plugins.form_elements.fields.textarea.apps module . . | 300 |
| 38.4.1.1.1.610 fobi.contrib.plugins.form_elements.fields.textarea.base module . . | 300 |
| 38.4.1.1.1.611 fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements module | 300 |
| 38.4.1.1.1.612 fobi.contrib.plugins.form_elements.fields.textarea.forms module . | 301 |
| 38.4.1.1.1.613 Module contents | 301 |
| 38.4.1.1.1.614 fobi.contrib.plugins.form_elements.fields.time package | 301 |
| 38.4.1.1.1.615 Submodules | 301 |
| 38.4.1.1.1.616 fobi.contrib.plugins.form_elements.fields.time.apps module | 301 |
| 38.4.1.1.1.617 fobi.contrib.plugins.form_elements.fields.time.base module | 302 |
| 38.4.1.1.1.618 fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements module | 302 |
| 38.4.1.1.1.619 fobi.contrib.plugins.form_elements.fields.time.forms module . . . | 303 |
| 38.4.1.1.1.620 Module contents | 303 |
| 38.4.1.1.1.621 fobi.contrib.plugins.form_elements.fields.url package | 303 |
| 38.4.1.1.1.622 Submodules | 303 |
| 38.4.1.1.1.623 fobi.contrib.plugins.form_elements.fields.url.apps module | 303 |
| 38.4.1.1.1.624 fobi.contrib.plugins.form_elements.fields.url.base module | 304 |
| 38.4.1.1.1.625 fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements module | 304 |
| 38.4.1.1.1.626 fobi.contrib.plugins.form_elements.fields.url.forms module | 305 |
| 38.4.1.1.1.627 Module contents | 305 |
| 38.4.1.1.1.628 Module contents | 305 |
| 38.4.1.1.1.629 fobi.contrib.plugins.form_elements.security package | 305 |
| 38.4.1.1.1.630 Subpackages | 305 |
| 38.4.1.1.1.631 fobi.contrib.plugins.form_elements.security.captcha package | 305 |
| 38.4.1.1.1.632 Submodules | 305 |
| 38.4.1.1.1.633 fobi.contrib.plugins.form_elements.security.captcha.apps module . | 305 |
| 38.4.1.1.1.634 fobi.contrib.plugins.form_elements.security.captcha.base module . | 306 |
| 38.4.1.1.1.635 fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements module | 306 |
| 38.4.1.1.1.636 fobi.contrib.plugins.form_elements.security.captcha.forms module | 307 |
| 38.4.1.1.1.637 Module contents | 307 |
| 38.4.1.1.1.638 fobi.contrib.plugins.form_elements.security.honeypot package . . | 307 |
| 38.4.1.1.1.639 Submodules | 307 |
| 38.4.1.1.1.640 fobi.contrib.plugins.form_elements.security.honeypot.apps module | 307 |
| 38.4.1.1.1.641 fobi.contrib.plugins.form_elements.security.honeypot.base module | 307 |
| 38.4.1.1.1.642 fobi.contrib.plugins.form_elements.security.honeypot.conf module | 308 |
| 38.4.1.1.1.643 fobi.contrib.plugins.form_elements.security.honeypot.defaults module | 308 |
| 38.4.1.1.1.644 fobi.contrib.plugins.form_elements.security.honeypot.fields module | 308 |
| 38.4.1.1.1.645 fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements module | 309 |
| 38.4.1.1.1.646 fobi.contrib.plugins.form_elements.security.honeypot.forms module | 309 |

| | |
|---|-----|
| 38.4.1.1.1.647 fobi.contrib.plugins.form_elements.security.honeypot.settings | |
| module | 309 |
| 38.4.1.1.1.648 Module contents | 310 |
| 38.4.1.1.1.649 fobi.contrib.plugins.form_elements.security.recaptcha package | 310 |
| 38.4.1.1.1.650 Submodules | 310 |
| 38.4.1.1.1.651 fobi.contrib.plugins.form_elements.security.recaptcha.apps module | 310 |
| 38.4.1.1.1.652 fobi.contrib.plugins.form_elements.security.recaptcha.base module | 310 |
| 38.4.1.1.1.653 fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements | |
| module | 310 |
| 38.4.1.1.1.654 fobi.contrib.plugins.form_elements.security.recaptcha.forms module | 311 |
| 38.4.1.1.1.655 Module contents | 311 |
| 38.4.1.1.1.656 Module contents | 311 |
| 38.4.1.1.1.657 fobi.contrib.plugins.form_elements.test package | 311 |
| 38.4.1.1.1.658 Subpackages | 311 |
| 38.4.1.1.1.659 fobi.contrib.plugins.form_elements.test.dummy package | 311 |
| 38.4.1.1.1.660 Submodules | 311 |
| 38.4.1.1.1.661 fobi.contrib.plugins.form_elements.test.dummy.apps module | 311 |
| 38.4.1.1.1.662 fobi.contrib.plugins.form_elements.test.dummy.base module | 312 |
| 38.4.1.1.1.663 fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements | |
| module | 312 |
| 38.4.1.1.1.664 fobi.contrib.plugins.form_elements.test.dummy.widgets module | 312 |
| 38.4.1.1.1.665 Module contents | 313 |
| 38.4.1.1.1.666 Module contents | 313 |
| 38.4.1.1.1.667 Module contents | 313 |
| 38.4.1.1.1.668 fobi.contrib.plugins.form_handlers package | 313 |
| 38.4.1.1.1.669 Subpackages | 313 |
| 38.4.1.1.1.670 fobi.contrib.plugins.form_handlers.db_store package | 313 |
| 38.4.1.1.1.671 Subpackages | 313 |
| 38.4.1.1.1.672 fobi.contrib.plugins.form_handlers.db_store.migrations package | 313 |
| 38.4.1.1.1.673 Submodules | 313 |
| 38.4.1.1.1.674 fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial | |
| module | 313 |
| 38.4.1.1.1.675 fobi.contrib.plugins.form_handlers.db_store.migrations.0002_savedformwizarddataentry | |
| module | 313 |
| 38.4.1.1.1.676 Module contents | 314 |
| 38.4.1.1.1.677 fobi.contrib.plugins.form_handlers.db_store.urls package | 314 |
| 38.4.1.1.1.678 Submodules | 314 |
| 38.4.1.1.1.679 fobi.contrib.plugins.form_handlers.db_store.urls.form_handlers | |
| module | 314 |
| 38.4.1.1.1.680 fobi.contrib.plugins.form_handlers.db_store.urls.form_wizard_handlers | |
| module | 314 |
| 38.4.1.1.1.681 Module contents | 314 |
| 38.4.1.1.1.682 Submodules | 314 |
| 38.4.1.1.1.683 fobi.contrib.plugins.form_handlers.db_store.admin module | 314 |
| 38.4.1.1.1.684 fobi.contrib.plugins.form_handlers.db_store.apps module | 315 |
| 38.4.1.1.1.685 fobi.contrib.plugins.form_handlers.db_store.base module | 315 |
| 38.4.1.1.1.686 fobi.contrib.plugins.form_handlers.db_store.callbacks module | 316 |
| 38.4.1.1.1.687 fobi.contrib.plugins.form_handlers.db_store.conf module | 317 |
| 38.4.1.1.1.688 fobi.contrib.plugins.form_handlers.db_store.defaults module | 317 |
| 38.4.1.1.1.689 fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers | |
| module | 317 |
| 38.4.1.1.1.690 fobi.contrib.plugins.form_handlers.db_store.helpers module | 318 |
| 38.4.1.1.1.691 fobi.contrib.plugins.form_handlers.db_store.models module | 319 |
| 38.4.1.1.1.692 fobi.contrib.plugins.form_handlers.db_store.settings module | 321 |

| | | |
|----------------|--|-----|
| 38.4.1.1.1.693 | fobi.contrib.plugins.form_handlers.db_store.views module | 321 |
| 38.4.1.1.1.694 | fobi.contrib.plugins.form_handlers.db_store.widgets module | 322 |
| 38.4.1.1.1.695 | Module contents | 322 |
| 38.4.1.1.1.696 | fobi.contrib.plugins.form_handlers.http_repost package | 322 |
| 38.4.1.1.1.697 | Submodules | 322 |
| 38.4.1.1.1.698 | fobi.contrib.plugins.form_handlers.http_repost.apps module | 322 |
| 38.4.1.1.1.699 | fobi.contrib.plugins.form_handlers.http_repost.base module | 323 |
| 38.4.1.1.1.700 | fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers module | 324 |
| 38.4.1.1.1.701 | fobi.contrib.plugins.form_handlers.http_repost.forms module | 325 |
| 38.4.1.1.1.702 | Module contents | 326 |
| 38.4.1.1.1.703 | fobi.contrib.plugins.form_handlers.mail package | 326 |
| 38.4.1.1.1.704 | Submodules | 326 |
| 38.4.1.1.1.705 | fobi.contrib.plugins.form_handlers.mail.apps module | 326 |
| 38.4.1.1.1.706 | fobi.contrib.plugins.form_handlers.mail.base module | 326 |
| 38.4.1.1.1.707 | fobi.contrib.plugins.form_handlers.mail.callbacks module | 327 |
| 38.4.1.1.1.708 | fobi.contrib.plugins.form_handlers.mail.conf module | 328 |
| 38.4.1.1.1.709 | fobi.contrib.plugins.form_handlers.mail.defaults module | 328 |
| 38.4.1.1.1.710 | fobi.contrib.plugins.form_handlers.mail.fields module | 328 |
| 38.4.1.1.1.711 | fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers module | 329 |
| 38.4.1.1.1.712 | fobi.contrib.plugins.form_handlers.mail.forms module | 330 |
| 38.4.1.1.1.713 | fobi.contrib.plugins.form_handlers.mail.helpers module | 330 |
| 38.4.1.1.1.714 | fobi.contrib.plugins.form_handlers.mail.mixins module | 331 |
| 38.4.1.1.1.715 | fobi.contrib.plugins.form_handlers.mail.settings module | 331 |
| 38.4.1.1.1.716 | fobi.contrib.plugins.form_handlers.mail.widgets module | 331 |
| 38.4.1.1.1.717 | Module contents | 332 |
| 38.4.1.1.1.718 | Module contents | 332 |
| 38.4.1.1.1.719 | fobi.contrib.plugins.form_importers package | 332 |
| 38.4.1.1.1.720 | Subpackages | 332 |
| 38.4.1.1.1.721 | fobi.contrib.plugins.form_importers.mailchimp_importer package | 332 |
| 38.4.1.1.1.722 | Submodules | 332 |
| 38.4.1.1.1.723 | fobi.contrib.plugins.form_importers.mailchimp_importer.apps module | 332 |
| 38.4.1.1.1.724 | fobi.contrib.plugins.form_importers.mailchimp_importer.base module | 332 |
| 38.4.1.1.1.725 | fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers module | 333 |
| 38.4.1.1.1.726 | fobi.contrib.plugins.form_importers.mailchimp_importer.forms module | 333 |
| 38.4.1.1.1.727 | fobi.contrib.plugins.form_importers.mailchimp_importer.views module | 334 |
| 38.4.1.1.1.728 | Module contents | 335 |
| 38.4.1.1.1.729 | Module contents | 335 |
| 38.4.1.1.1.730 | Module contents | 335 |
| 38.4.1.1.1.731 | fobi.contrib.themes package | 335 |
| 38.4.1.1.1.732 | Subpackages | 335 |
| 38.4.1.1.1.733 | fobi.contrib.themes.bootstrap3 package | 335 |
| 38.4.1.1.1.734 | Subpackages | 335 |
| 38.4.1.1.1.735 | fobi.contrib.themes.bootstrap3.widgets package | 335 |
| 38.4.1.1.1.736 | Subpackages | 335 |
| 38.4.1.1.1.737 | fobi.contrib.themes.bootstrap3.widgets.form_elements package | 335 |
| 38.4.1.1.1.738 | Subpackages | 335 |
| 38.4.1.1.1.739 | fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_wi package | 335 |

| | |
|---|-----|
| 38.4.1.1.1.740 Submodules | 335 |
| 38.4.1.1.1.741 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget module | 335 |
| 38.4.1.1.1.742 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget module | 335 |
| 38.4.1.1.1.743 Module contents | 336 |
| 38.4.1.1.1.744 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget package | 336 |
| 38.4.1.1.1.745 Submodules | 336 |
| 38.4.1.1.1.746 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget module | 336 |
| 38.4.1.1.1.747 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget module | 336 |
| 38.4.1.1.1.748 Module contents | 336 |
| 38.4.1.1.1.749 fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget package | 336 |
| 38.4.1.1.1.750 Submodules | 336 |
| 38.4.1.1.1.751 fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps module | 336 |
| 38.4.1.1.1.752 fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form module | 337 |
| 38.4.1.1.1.753 Module contents | 337 |
| 38.4.1.1.1.754 fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget package | 337 |
| 38.4.1.1.1.755 Submodules | 337 |
| 38.4.1.1.1.756 fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps module | 337 |
| 38.4.1.1.1.757 fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_f module | 337 |
| 38.4.1.1.1.758 Module contents | 338 |
| 38.4.1.1.1.759 fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget package | 338 |
| 38.4.1.1.1.760 Submodules | 338 |
| 38.4.1.1.1.761 fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps module | 338 |
| 38.4.1.1.1.762 fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_fo module | 338 |
| 38.4.1.1.1.763 Module contents | 338 |
| 38.4.1.1.1.764 fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget package | 338 |
| 38.4.1.1.1.765 Submodules | 338 |
| 38.4.1.1.1.766 fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps module | 338 |
| 38.4.1.1.1.767 fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_for module | 339 |
| 38.4.1.1.1.768 Module contents | 339 |
| 38.4.1.1.1.769 Module contents | 339 |
| 38.4.1.1.1.770 Module contents | 339 |
| 38.4.1.1.1.771 Submodules | 339 |
| 38.4.1.1.1.772 fobi.contrib.themes.bootstrap3.apps module | 339 |
| 38.4.1.1.1.773 fobi.contrib.themes.bootstrap3.fobi_themes module | 339 |
| 38.4.1.1.1.774 Module contents | 341 |
| 38.4.1.1.1.775 fobi.contrib.themes.djangocms_admin_style_theme package | 341 |
| 38.4.1.1.1.776 Subpackages | 341 |

| | | |
|----------------|--|-----|
| 38.4.1.1.1.777 | fobi.contrib.themes.djangocms_admin_style_theme.widgets package | 341 |
| 38.4.1.1.1.778 | Subpackages | 341 |
| 38.4.1.1.1.779 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements package | 341 |
| 38.4.1.1.1.780 | Subpackages | 341 |
| 38.4.1.1.1.781 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext package | 341 |
| 38.4.1.1.1.782 | Submodules | 341 |
| 38.4.1.1.1.783 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext module | 341 |
| 38.4.1.1.1.784 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext module | 342 |
| 38.4.1.1.1.785 | Module contents | 342 |
| 38.4.1.1.1.786 | Module contents | 342 |
| 38.4.1.1.1.787 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers package | 342 |
| 38.4.1.1.1.788 | Subpackages | 342 |
| 38.4.1.1.1.789 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store package | 342 |
| 38.4.1.1.1.790 | Submodules | 342 |
| 38.4.1.1.1.791 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps module | 342 |
| 38.4.1.1.1.792 | fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi module | 342 |
| 38.4.1.1.1.793 | Module contents | 343 |
| 38.4.1.1.1.794 | Module contents | 343 |
| 38.4.1.1.1.795 | Module contents | 343 |
| 38.4.1.1.1.796 | Submodules | 343 |
| 38.4.1.1.1.797 | fobi.contrib.themes.djangocms_admin_style_theme.apps module | 343 |
| 38.4.1.1.1.798 | fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes module | 343 |
| 38.4.1.1.1.799 | Module contents | 345 |
| 38.4.1.1.1.800 | fobi.contrib.themes.foundation5 package | 345 |
| 38.4.1.1.1.801 | Subpackages | 345 |
| 38.4.1.1.1.802 | fobi.contrib.themes.foundation5.widgets package | 345 |
| 38.4.1.1.1.803 | Subpackages | 345 |
| 38.4.1.1.1.804 | fobi.contrib.themes.foundation5.widgets.form_elements package | 345 |
| 38.4.1.1.1.805 | Subpackages | 345 |
| 38.4.1.1.1.806 | fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widgets package | 345 |
| 38.4.1.1.1.807 | Submodules | 345 |
| 38.4.1.1.1.808 | fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widgets module | 345 |
| 38.4.1.1.1.809 | fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widgets module | 345 |
| 38.4.1.1.1.810 | Module contents | 346 |
| 38.4.1.1.1.811 | fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget package | 346 |
| 38.4.1.1.1.812 | Submodules | 346 |
| 38.4.1.1.1.813 | fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps module | 346 |
| 38.4.1.1.1.814 | fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_for module | 346 |

| | | |
|----------------|--|-----|
| 38.4.1.1.1.815 | Module contents | 346 |
| 38.4.1.1.1.816 | fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget package | 346 |
| 38.4.1.1.1.817 | Submodules | 346 |
| 38.4.1.1.1.818 | fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps module | 346 |
| 38.4.1.1.1.819 | fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.fobi module | 347 |
| 38.4.1.1.1.820 | Module contents | 347 |
| 38.4.1.1.1.821 | fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget package | 347 |
| 38.4.1.1.1.822 | Submodules | 347 |
| 38.4.1.1.1.823 | fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps module | 347 |
| 38.4.1.1.1.824 | fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi module | 347 |
| 38.4.1.1.1.825 | Module contents | 348 |
| 38.4.1.1.1.826 | Module contents | 348 |
| 38.4.1.1.1.827 | fobi.contrib.themes.foundation5.widgets.form_handlers package | 348 |
| 38.4.1.1.1.828 | Subpackages | 348 |
| 38.4.1.1.1.829 | fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget package | 348 |
| 38.4.1.1.1.830 | Submodules | 348 |
| 38.4.1.1.1.831 | fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps module | 348 |
| 38.4.1.1.1.832 | fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi module | 348 |
| 38.4.1.1.1.833 | Module contents | 349 |
| 38.4.1.1.1.834 | Module contents | 349 |
| 38.4.1.1.1.835 | Module contents | 349 |
| 38.4.1.1.1.836 | Submodules | 349 |
| 38.4.1.1.1.837 | fobi.contrib.themes.foundation5.apps module | 349 |
| 38.4.1.1.1.838 | fobi.contrib.themes.foundation5.fobi_themes module | 349 |
| 38.4.1.1.1.839 | Module contents | 351 |
| 38.4.1.1.1.840 | fobi.contrib.themes.simple package | 351 |
| 38.4.1.1.1.841 | Subpackages | 351 |
| 38.4.1.1.1.842 | fobi.contrib.themes.simple.widgets package | 351 |
| 38.4.1.1.1.843 | Subpackages | 351 |
| 38.4.1.1.1.844 | fobi.contrib.themes.simple.widgets.form_elements package | 351 |
| 38.4.1.1.1.845 | Subpackages | 351 |
| 38.4.1.1.1.846 | fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget package | 351 |
| 38.4.1.1.1.847 | Submodules | 351 |
| 38.4.1.1.1.848 | fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.apps module | 351 |
| 38.4.1.1.1.849 | fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi_f module | 351 |
| 38.4.1.1.1.850 | Module contents | 352 |
| 38.4.1.1.1.851 | Module contents | 352 |
| 38.4.1.1.1.852 | fobi.contrib.themes.simple.widgets.form_handlers package | 352 |
| 38.4.1.1.1.853 | Subpackages | 352 |
| 38.4.1.1.1.854 | fobi.contrib.themes.simple.widgets.form_handlers.db_store package | 352 |
| 38.4.1.1.1.855 | Submodules | 352 |

| | | |
|----------------|---|-----|
| 38.4.1.1.1.856 | fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps module | 352 |
| 38.4.1.1.1.857 | fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements module | 352 |
| 38.4.1.1.1.858 | Module contents | 352 |
| 38.4.1.1.1.859 | Module contents | 352 |
| 38.4.1.1.1.860 | Module contents | 352 |
| 38.4.1.1.1.861 | Submodules | 352 |
| 38.4.1.1.1.862 | fobi.contrib.themes.simple.apps module | 352 |
| 38.4.1.1.1.863 | fobi.contrib.themes.simple.fobi_themes module | 353 |
| 38.4.1.1.1.864 | Module contents | 354 |
| 38.4.1.1.1.865 | Module contents | 354 |
| 38.4.1.1.2 | Module contents | 354 |
| 38.4.1.2 | fobi.integration package | 354 |
| 38.4.1.2.1 | Submodules | 354 |
| 38.4.1.2.2 | fobi.integration.helpers module | 354 |
| 38.4.1.2.3 | fobi.integration.processors module | 355 |
| 38.4.1.2.4 | Module contents | 356 |
| 38.4.1.3 | fobi.management package | 356 |
| 38.4.1.3.1 | Subpackages | 356 |
| 38.4.1.3.1.1 | fobi.management.commands package | 356 |
| 38.4.1.3.1.2 | Submodules | 356 |
| 38.4.1.3.1.3 | fobi.management.commands.fobi_find_broken_entries module | 356 |
| 38.4.1.3.1.4 | fobi.management.commands.fobi_sync_plugins module | 356 |
| 38.4.1.3.1.5 | fobi.management.commands.fobi_update_plugin_data module | 356 |
| 38.4.1.3.1.6 | Module contents | 357 |
| 38.4.1.3.2 | Module contents | 357 |
| 38.4.1.4 | fobi.migrations package | 357 |
| 38.4.1.4.1 | Submodules | 357 |
| 38.4.1.4.2 | fobi.migrations.0001_initial module | 357 |
| 38.4.1.4.3 | fobi.migrations.0002_auto_20150912_1744 module | 357 |
| 38.4.1.4.4 | fobi.migrations.0003_auto_20160517_1005 module | 357 |
| 38.4.1.4.5 | fobi.migrations.0004_auto_20160906_1513 module | 357 |
| 38.4.1.4.6 | fobi.migrations.0005_auto_20160908_1457 module | 358 |
| 38.4.1.4.7 | fobi.migrations.0006_auto_20160911_1549 module | 358 |
| 38.4.1.4.8 | fobi.migrations.0007_auto_20160926_1652 module | 358 |
| 38.4.1.4.9 | fobi.migrations.0008_formwizardhandlerentry module | 358 |
| 38.4.1.4.10 | fobi.migrations.0009_formwizardentry_wizard_type module | 358 |
| 38.4.1.4.11 | fobi.migrations.0010_formwizardhandler module | 358 |
| 38.4.1.4.12 | fobi.migrations.0011_formentry_title module | 359 |
| 38.4.1.4.13 | fobi.migrations.0012_auto_20161109_1550 module | 359 |
| 38.4.1.4.14 | fobi.migrations.0013_formwizardentry_show_all_navigation_buttons module | 359 |
| 38.4.1.4.15 | fobi.migrations.0014_auto_20170516_1413 module | 359 |
| 38.4.1.4.16 | Module contents | 359 |
| 38.4.1.5 | fobi.reusable package | 359 |
| 38.4.1.5.1 | Subpackages | 359 |
| 38.4.1.5.1.1 | fobi.reusable.markdown_widget package | 359 |
| 38.4.1.5.1.2 | Submodules | 359 |
| 38.4.1.5.1.3 | fobi.reusable.markdown_widget.helpers module | 359 |
| 38.4.1.5.1.4 | fobi.reusable.markdown_widget.widgets module | 360 |
| 38.4.1.5.1.5 | Module contents | 360 |
| 38.4.1.5.2 | Module contents | 360 |
| 38.4.1.6 | fobi.templatetags package | 360 |

| | | |
|--------------|--|-----|
| 38.4.1.6.1 | Submodules | 360 |
| 38.4.1.6.2 | fobi.templatetags.fobi_tags module | 360 |
| 38.4.1.6.3 | fobi.templatetags.future_compat module | 362 |
| 38.4.1.6.4 | Module contents | 362 |
| 38.4.1.7 | fobi.tests package | 362 |
| 38.4.1.7.1 | Submodules | 362 |
| 38.4.1.7.2 | fobi.tests.base module | 362 |
| 38.4.1.7.3 | fobi.tests.constants module | 363 |
| 38.4.1.7.4 | fobi.tests.core module | 363 |
| 38.4.1.7.5 | fobi.tests.data module | 363 |
| 38.4.1.7.6 | fobi.tests.helpers module | 363 |
| 38.4.1.7.7 | fobi.tests.test_browser_build_dynamic_forms module | 364 |
| 38.4.1.7.8 | fobi.tests.test_core module | 365 |
| 38.4.1.7.9 | fobi.tests.test_drf_integration module | 365 |
| 38.4.1.7.10 | fobi.tests.test_dynamic_forms module | 366 |
| 38.4.1.7.11 | fobi.tests.test_feincms_integration module | 366 |
| 38.4.1.7.12 | fobi.tests.test_form_importers_mailchimp module | 366 |
| 38.4.1.7.13 | fobi.tests.test_sortable_dict module | 367 |
| 38.4.1.7.14 | Module contents | 367 |
| 38.4.1.8 | fobi.urls package | 367 |
| 38.4.1.8.1 | Submodules | 367 |
| 38.4.1.8.2 | fobi.urls.edit module | 367 |
| 38.4.1.8.3 | fobi.urls.view module | 367 |
| 38.4.1.8.4 | Module contents | 367 |
| 38.4.1.9 | fobi.wizard package | 367 |
| 38.4.1.9.1 | Subpackages | 367 |
| 38.4.1.9.1.1 | fobi.wizard.views package | 367 |
| 38.4.1.9.1.2 | Submodules | 367 |
| 38.4.1.9.1.3 | fobi.wizard.views.dynamic module | 367 |
| 38.4.1.9.1.4 | fobi.wizard.views.views module | 372 |
| 38.4.1.9.1.5 | Module contents | 373 |
| 38.4.1.9.2 | Module contents | 373 |
| 38.4.2 | Submodules | 373 |
| 38.4.3 | fobi.admin module | 373 |
| 38.4.4 | fobi.app module | 377 |
| 38.4.5 | fobi.apps module | 378 |
| 38.4.6 | fobi.base module | 378 |
| 38.4.7 | fobi.conf module | 397 |
| 38.4.8 | fobi.constants module | 397 |
| 38.4.9 | fobi.context_processors module | 397 |
| 38.4.10 | fobi.data_structures module | 398 |
| 38.4.11 | fobi.decorators module | 399 |
| 38.4.12 | fobi.defaults module | 400 |
| 38.4.13 | fobi.discover module | 400 |
| 38.4.14 | fobi.dynamic module | 401 |
| 38.4.15 | fobi.exceptions module | 401 |
| 38.4.16 | fobi.form_importers module | 403 |
| 38.4.17 | fobi.form_utils module | 404 |
| 38.4.18 | fobi.forms module | 404 |
| 38.4.19 | fobi.helpers module | 409 |
| 38.4.20 | fobi.models module | 414 |
| 38.4.21 | fobi.settings module | 429 |
| 38.4.22 | fobi.test module | 429 |
| 38.4.23 | fobi.utils module | 429 |

| | | |
|------------|---|------------|
| 38.4.24 | fobi.validators module | 432 |
| 38.4.25 | fobi.views module | 432 |
| 38.4.26 | fobi.widgets module | 438 |
| 38.4.27 | Module contents | 438 |
| 39 | Indices and tables | 439 |
| 40 | (Sub)modules | 441 |
| 40.1 | fobi.contrib.apps.djangocms_integration | 441 |
| 40.1.1 | Prerequisites | 441 |
| 40.1.2 | Installation | 441 |
| 40.1.2.1 | Versions | 441 |
| 40.1.2.2 | your_project/settings.py | 441 |
| 40.1.3 | Information for developers | 442 |
| 40.1.3.1 | Templates for DjangoCMS | 442 |
| 40.1.3.2 | Template rendering | 442 |
| 40.1.3.3 | Using custom templates for rendering the widget | 442 |
| 40.1.4 | Usage | 443 |
| 40.2 | fobi.contrib.apps.drf_integration | 443 |
| 40.2.1 | Live demo | 444 |
| 40.2.2 | Supported fields | 444 |
| 40.2.2.1 | Content (presentational form elements) | 444 |
| 40.2.2.2 | Fields | 444 |
| 40.2.3 | Not (yet) supported fields | 445 |
| 40.2.4 | Implementation details | 445 |
| 40.2.5 | Installation | 448 |
| 40.2.5.1 | Versions | 448 |
| 40.2.5.2 | your_project/settings.py | 448 |
| 40.2.5.3 | your_project/urls.py | 449 |
| 40.2.6 | Usage | 449 |
| 40.2.6.1 | LIST | 449 |
| 40.2.6.2 | OPTIONS | 449 |
| 40.2.6.3 | PUT | 453 |
| 40.2.7 | Callbacks | 453 |
| 40.2.7.1 | fobi_form_callbacks.py | 453 |
| 40.2.8 | Testing | 454 |
| 40.2.9 | Limitations | 454 |
| 40.3 | Form elements | 454 |
| 40.3.1 | Content form element | 454 |
| 40.3.1.1 | fobi.contrib.apps.drf_integration.form_elements.content.content_image | 454 |
| 40.3.1.1.1 | Installation | 454 |
| 40.3.1.1.2 | Usage | 455 |
| 40.3.1.2 | fobi.contrib.apps.drf_integration.form_elements.content.content_image_url | 455 |
| 40.3.1.2.1 | Installation | 456 |
| 40.3.1.2.2 | Usage | 456 |
| 40.3.1.3 | fobi.contrib.apps.drf_integration.form_elements.content.content_richtext | 457 |
| 40.3.1.3.1 | Installation | 457 |
| 40.3.1.3.2 | Usage | 457 |
| 40.3.1.4 | fobi.contrib.apps.drf_integration.form_elements.content.content_text | 458 |
| 40.3.1.4.1 | Installation | 458 |
| 40.3.1.4.2 | Usage | 458 |
| 40.3.1.5 | fobi.contrib.apps.drf_integration.form_elements.content.content_video | 459 |
| 40.3.1.5.1 | Installation | 459 |
| 40.3.1.5.2 | Usage | 459 |

| | | |
|-------------|---|-----|
| 40.3.2 | Fields | 460 |
| 40.3.2.1 | fobi.contrib.apps.drf_integration.form_elements.fields.boolean | 460 |
| 40.3.2.1.1 | Installation | 460 |
| 40.3.2.2 | fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple | 460 |
| 40.3.2.2.1 | Installation | 460 |
| 40.3.2.3 | fobi.contrib.apps.drf_integration.form_elements.fields.date | 461 |
| 40.3.2.3.1 | Installation | 461 |
| 40.3.2.4 | fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down | 461 |
| 40.3.2.4.1 | Installation | 461 |
| 40.3.2.5 | fobi.contrib.apps.drf_integration.form_elements.fields.datetime | 462 |
| 40.3.2.5.1 | Installation | 462 |
| 40.3.2.6 | fobi.contrib.apps.drf_integration.form_elements.fields.decimal | 462 |
| 40.3.2.6.1 | Installation | 462 |
| 40.3.2.7 | fobi.contrib.apps.drf_integration.form_elements.fields.email | 462 |
| 40.3.2.7.1 | Installation | 463 |
| 40.3.2.8 | fobi.contrib.apps.drf_integration.form_elements.fields.file | 463 |
| 40.3.2.8.1 | Installation | 463 |
| 40.3.2.9 | fobi.contrib.apps.drf_integration.form_elements.fields.float | 463 |
| 40.3.2.9.1 | Installation | 463 |
| 40.3.2.10 | fobi.contrib.apps.drf_integration.form_elements.fields.hidden | 464 |
| 40.3.2.10.1 | Installation | 464 |
| 40.3.2.11 | fobi.contrib.apps.drf_integration.form_elements.fields.input | 464 |
| 40.3.2.11.1 | Installation | 464 |
| 40.3.2.12 | fobi.contrib.apps.drf_integration.form_elements.fields.integer | 465 |
| 40.3.2.12.1 | Installation | 465 |
| 40.3.2.13 | fobi.contrib.apps.drf_integration.form_elements.fields.ip_address | 465 |
| 40.3.2.13.1 | Installation | 465 |
| 40.3.2.14 | fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean | 466 |
| 40.3.2.14.1 | Installation | 466 |
| 40.3.2.15 | fobi.contrib.apps.drf_integration.form_elements.fields.password | 466 |
| 40.3.2.15.1 | Installation | 466 |
| 40.3.2.16 | fobi.contrib.apps.drf_integration.form_elements.fields.radio | 467 |
| 40.3.2.16.1 | Installation | 467 |
| 40.3.2.17 | fobi.contrib.apps.drf_integration.form_elements.fields.range_select | 467 |
| 40.3.2.17.1 | Installation | 467 |
| 40.3.2.18 | fobi.contrib.apps.drf_integration.form_elements.fields.regex | 467 |
| 40.3.2.18.1 | Installation | 468 |
| 40.3.2.19 | fobi.contrib.apps.drf_integration.form_elements.fields.select | 468 |
| 40.3.2.19.1 | Installation | 468 |
| 40.3.2.20 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple | 468 |
| 40.3.2.20.1 | Installation | 468 |
| 40.3.2.21 | fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple | 469 |
| 40.3.2.21.1 | Installation | 469 |
| 40.3.2.22 | fobi.contrib.apps.drf_integration.form_elements.fields.slider | 469 |
| 40.3.2.22.1 | Installation | 469 |
| 40.3.2.23 | fobi.contrib.apps.drf_integration.form_elements.fields.slug | 470 |
| 40.3.2.23.1 | Installation | 470 |
| 40.3.2.24 | fobi.contrib.apps.drf_integration.form_elements.fields.text | 470 |
| 40.3.2.24.1 | Installation | 470 |
| 40.3.2.25 | fobi.contrib.apps.drf_integration.form_elements.fields.textarea | 471 |
| 40.3.2.25.1 | Installation | 471 |
| 40.3.2.26 | fobi.contrib.apps.drf_integration.form_elements.fields.time | 471 |
| 40.3.2.26.1 | Installation | 471 |
| 40.3.2.27 | fobi.contrib.apps.drf_integration.form_elements.fields.url | 471 |

| | | |
|-------------|--|-----|
| 40.3.2.27.1 | Installation | 472 |
| 40.4 | Form handlers | 472 |
| 40.4.1 | fobi.contrib.apps.drf_integration.form_handlers.db_store | 472 |
| 40.4.1.1 | Installation | 472 |
| 40.4.2 | fobi.contrib.apps.drf_integration.form_handlers.http_repost | 472 |
| 40.4.2.1 | Installation | 473 |
| 40.4.3 | fobi.contrib.apps.drf_integration.form_handlers.mail | 473 |
| 40.4.3.1 | Installation | 473 |
| 40.5 | fobi.contrib.apps.feincms_integration | 473 |
| 40.5.1 | Prerequisites | 473 |
| 40.5.2 | Installation | 474 |
| 40.5.2.1 | Versions | 474 |
| 40.5.2.2 | your_project/settings.py | 474 |
| 40.5.2.3 | your_project/page/models.py | 474 |
| 40.5.2.4 | your_project/admin.py | 475 |
| 40.5.2.5 | Information for developers | 475 |
| 40.5.2.5.1 | Template rendering | 475 |
| 40.5.2.6 | Using custom templates for rendering the widget | 475 |
| 40.5.3 | Usage | 476 |
| 40.6 | fobi.contrib.apps.mezzanine_integration | 476 |
| 40.6.1 | Prerequisites | 477 |
| 40.6.2 | Installation | 477 |
| 40.6.2.1 | Versions | 477 |
| 40.6.2.2 | your_project/settings.py | 477 |
| 40.6.3 | Information for developers | 477 |
| 40.6.3.1 | Template rendering | 477 |
| 40.6.3.2 | Using custom templates for rendering the form | 477 |
| 40.6.4 | Usage | 478 |
| 40.6.4.1 | The <code>fobi</code> page model | 478 |
| 40.6.4.2 | Steps described | 478 |
| 40.7 | fobi.contrib.plugins.form_elements.content.content_image | 479 |
| 40.7.1 | Installation | 479 |
| 40.8 | fobi.contrib.plugins.form_elements.content.content_image_url | 479 |
| 40.8.1 | Installation | 479 |
| 40.9 | fobi.contrib.plugins.form_elements.content.content_text | 480 |
| 40.9.1 | Installation | 480 |
| 40.10 | fobi.contrib.plugins.form_elements.content.content_richtext | 481 |
| 40.10.1 | Installation | 481 |
| 40.10.2 | Controlling HTML tags and attributes | 482 |
| 40.11 | fobi.contrib.plugins.form_elements.content.content_video | 482 |
| 40.11.1 | Installation | 482 |
| 40.12 | fobi.contrib.plugins.form_elements.fields.boolean | 483 |
| 40.12.1 | Installation | 483 |
| 40.13 | fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple | 483 |
| 40.13.1 | Installation | 483 |
| 40.13.2 | Usage | 484 |
| 40.14 | fobi.contrib.plugins.form_elements.fields.date | 485 |
| 40.14.1 | Installation | 485 |
| 40.15 | fobi.contrib.plugins.form_elements.fields.date_drop_down | 485 |
| 40.15.1 | Installation | 485 |
| 40.16 | fobi.contrib.plugins.form_elements.fields.datetime | 486 |
| 40.16.1 | Installation | 486 |
| 40.17 | fobi.contrib.plugins.form_elements.fields.decimal | 486 |
| 40.17.1 | Installation | 486 |

| | | |
|-----------|--|-----|
| 40.18 | fobi.contrib.plugins.form_elements.fields.email | 487 |
| 40.18.1 | Installation | 487 |
| 40.19 | fobi.contrib.plugins.form_elements.fields.file | 487 |
| 40.19.1 | Installation | 487 |
| 40.20 | fobi.contrib.plugins.form_elements.fields.float | 488 |
| 40.20.1 | Installation | 488 |
| 40.21 | fobi.contrib.plugins.form_elements.fields.hidden | 488 |
| 40.21.1 | Installation | 488 |
| 40.22 | fobi.contrib.plugins.form_elements.fields.input | 489 |
| 40.22.1 | Installation | 489 |
| 40.23 | fobi.contrib.plugins.form_elements.fields.integer | 490 |
| 40.23.1 | Installation | 490 |
| 40.24 | fobi.contrib.plugins.form_elements.fields.ip_address | 490 |
| 40.24.1 | Installation | 490 |
| 40.25 | fobi.contrib.plugins.form_elements.fields.null_boolean | 491 |
| 40.25.1 | Installation | 491 |
| 40.26 | fobi.contrib.plugins.form_elements.fields.password | 491 |
| 40.26.1 | Installation | 491 |
| 40.27 | fobi.contrib.plugins.form_elements.fields.radio | 491 |
| 40.27.1 | Installation | 492 |
| 40.27.2 | Usage | 492 |
| 40.28 | fobi.contrib.plugins.form_elements.fields.range_select | 493 |
| 40.28.1 | Installation | 493 |
| 40.29 | fobi.contrib.plugins.form_elements.fields.regex | 494 |
| 40.29.1 | Installation | 494 |
| 40.30 | fobi.contrib.plugins.form_elements.fields.select | 494 |
| 40.30.1 | Installation | 494 |
| 40.30.2 | Usage | 495 |
| 40.31 | fobi.contrib.plugins.form_elements.fields.select_model_object | 496 |
| 40.31.1 | Installation | 496 |
| 40.32 | fobi.contrib.plugins.form_elements.fields.select_mptt_model_object | 496 |
| 40.32.1 | Installation | 496 |
| 40.32.1.1 | Install <i>django-mptt</i> | 496 |
| 40.32.1.2 | Install <i>select_mptt_model_object</i> plugin | 497 |
| 40.33 | fobi.contrib.plugins.form_elements.fields.select_multiple | 497 |
| 40.33.1 | Installation | 497 |
| 40.33.2 | Usage | 498 |
| 40.34 | fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects | 499 |
| 40.34.1 | Installation | 499 |
| 40.35 | fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects | 499 |
| 40.35.1 | Installation | 499 |
| 40.35.1.1 | Install <i>django-mptt</i> | 499 |
| 40.35.1.2 | Install <i>select_multiple_mptt_model_objects</i> plugin | 500 |
| 40.36 | fobi.contrib.plugins.form_elements.fields.select_multiple_with_max | 500 |
| 40.36.1 | Installation | 501 |
| 40.36.2 | Usage | 501 |
| 40.37 | fobi.contrib.plugins.form_elements.fields.slider | 502 |
| 40.37.1 | Installation | 502 |
| 40.38 | fobi.contrib.plugins.form_elements.fields.text | 503 |
| 40.38.1 | Installation | 503 |
| 40.39 | fobi.contrib.plugins.form_elements.fields.textarea | 503 |
| 40.39.1 | Installation | 503 |
| 40.40 | fobi.contrib.plugins.form_elements.fields.time | 504 |
| 40.40.1 | Installation | 504 |

| | | |
|----------------------------|--|------------|
| 40.41 | fobi.contrib.plugins.form_elements.fields.url | 504 |
| 40.41.1 | Installation | 504 |
| 40.42 | fobi.contrib.plugins.form_elements.security.captcha | 505 |
| 40.42.1 | Prerequisites | 505 |
| 40.42.2 | Installation | 505 |
| 40.42.2.1 | Install <i>django-simple-captcha</i> | 505 |
| 40.42.2.2 | Install <i>fobi</i> Captcha plugin | 505 |
| 40.42.3 | Troubleshooting and usage limitations | 506 |
| 40.42.3.1 | In combination with other captcha solutions | 506 |
| 40.42.4 | Usage | 506 |
| 40.43 | fobi.contrib.plugins.form_elements.security.honeypot | 506 |
| 40.43.1 | Installation | 506 |
| 40.44 | fobi.contrib.plugins.form_elements.security.recaptcha | 506 |
| 40.44.1 | Installation | 507 |
| 40.44.1.1 | Install <i>django-recaptcha</i> | 507 |
| 40.44.1.2 | Install <i>fobi</i> ReCAPTCHA plugin | 507 |
| 40.44.2 | Troubleshooting and usage limitations | 507 |
| 40.44.2.1 | In combination with other captcha solutions | 507 |
| 40.44.3 | Usage | 507 |
| 40.45 | fobi.contrib.plugins.form_elements.test.dummy | 508 |
| 40.45.1 | Installation | 508 |
| 40.46 | fobi.contrib.plugins.form_handlers.db_store | 508 |
| 40.46.1 | Dependencies | 508 |
| 40.46.2 | Installation | 508 |
| 40.47 | fobi.contrib.plugins.form_handlers.http_repost | 509 |
| 40.47.1 | Installation | 509 |
| 40.48 | fobi.contrib.plugins.form_handlers.mail | 509 |
| 40.48.1 | Installation | 509 |
| 40.49 | mailchimp_importer | 510 |
| 40.49.1 | Prerequisites | 510 |
| 40.49.2 | Installation | 510 |
| 40.49.2.1 | your_project/settings.py | 510 |
| 40.49.3 | How it works | 510 |
| 40.49.4 | Development status | 511 |
| 40.50 | fobi.contrib.themes.bootstrap3 | 511 |
| 40.50.1 | Installation | 511 |
| 40.51 | fobi.contrib.themes.djangocms_admin_style_theme | 511 |
| 40.51.1 | Installation | 511 |
| 40.51.1.1 | Install <i>djangocms-admin-style</i> | 511 |
| 40.51.1.2 | Install <i>fobi.contrib.themes.djangocms_admin_style_theme</i> theme | 511 |
| 40.52 | fobi.contrib.themes.foundation5 | 512 |
| 40.52.1 | Installation | 512 |
| 40.53 | fobi.contrib.themes.simple | 512 |
| 40.53.1 | Installation | 512 |
| Bibliography | | 513 |
| Python Module Index | | 515 |
| Index | | 527 |

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).

CHAPTER 1

Prerequisites

- Django 2.2, 3.0 and 3.1.
- Python 3.5, 3.6, 3.7, 3.8 and 3.9.

CHAPTER 2

Key concepts

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

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

Main features and highlights

- User-friendly GUI to quickly build forms.
- Large variety of *Bundled form element plugins*. Most of the Django fields are supported. *HTML5 fields* are supported as well.
- *Form wizards*. Combine your forms into wizards. Form wizards may contain handlers. Handler processes the form wizard data (for example, saves it or mails it). Number of the form wizard handlers is not limited.
- Forms can be automatically enabled/disabled based on dates (start date, end date).
- Anti-spam solutions like *CAPTCHA*, *ReCAPTCHA*, *Honeypot* or *Invisible reCAPTCHA* come out of the box (CAPTCHA and ReCAPTCHA do require additional third-party apps to be installed; Invisible reCAPTCHA doesn't).
- In addition to standard form elements, there are cosmetic (presentational) form elements (for adding a piece of text, image or a embed video) alongside standard form elements.
- Data handling in plugins (form handlers). Save the data, mail it to some address or re-post it to some other endpoint. See the *Bundled form handler plugins* for more information.
- Developer-friendly API, which allows to edit existing or build new form fields and handlers without touching the core.
- Support for custom user model.
- *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 Django REST framework*.
- Implemented *integration with Wagtail* (in a form of a Wagtail page).
- Implemented *integration with FeinCMS* (in a form of a FeinCMS page widget).
- Implemented *integration with DjangoCMS* (in a form of a DjangoCMS page plugin).
- Implemented *integration with Mezzanine* (in a form of a Mezzanine page).
- Reordering of form elements using drag-n-drop.

- Data export ([DB store](#) form handler plugin) into XLS/CSV format.
- *Dynamic initial values* for form elements.
- Import/export forms to/from JSON format.
- Import forms from MailChimp using [mailchimp importer](#).

CHAPTER 4

Roadmap

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

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

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

CHAPTER 5

Some screenshots

See the documentation for some screen shots:

- [ReadTheDocs](#)

6.1 Live demo

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

Credentials:

- username: test_user
- password: test_user

6.2 Run demo locally

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

Grab the latest `django_fobi_example_app_installer.sh`:

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

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

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

Open your browser and test the app.

Dashboard:

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

- Admin password: test

Django admin interface:

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

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

CHAPTER 7

Quick start

See the [quick start](#).

Installation

(1) Install latest stable version from PyPI:

```
pip install django-fobi
```

Or latest stable version from GitHub:

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

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

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

(continues on next page)

(continued from previous page)

```

'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_image_url',
'fobi.contrib.plugins.form_elements.content.content_text',
'fobi.contrib.plugins.form_elements.content.content_video',

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

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

```

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

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

```

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [(os.path.join('path', 'to', 'your', 'templates'))],
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',

```

(continues on next page)

(continued from previous page)

```

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

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

(4) Configure URLs

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

```

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

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

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

```

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

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

Creating a new form element plugin

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

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

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

You should see a form element plugin as a Django micro app, which could have its' own models, admin interface, etc. *django-fobi* comes with several bundled form element plugins. Do check the source code as example.

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

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

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

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

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

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

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

9.1 Define and register the form element plugin

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

9.1.1 path/to/sample_textarea/fobi_form_elements.py

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

Required imports.

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

Defining the Sample textarea plugin.

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

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

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

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

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

Registering the SampleTextareaPlugin plugin.

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

Note, that in case you want to define a pure presentational element, make use of `fobi.base.FormElementPlugin` for subclassing, instead of `fobi.base.FormFieldPlugin`. See the source of the content plugins (`fobi.contrib.plugins.form_elements.content`) as a 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 `django.forms.widgets.TextInput` widget in edit mode. In the view mode, the original widget that you assigned in your form element plugin would be used.

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

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

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

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

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

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

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

9.1.2 path/to/sample_textarea/forms.py

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

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

Required imports.

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

Form for for `SampleTextareaPlugin` form element plugin.

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

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

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

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

- name

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

See the following example.

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

9.1.3 path/to/sample_textarea/widgets.py

Required imports.

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

Defining the base plugin widget.

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

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

9.1.4 path/to/sample_layout/fobi_form_elements.py

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

Required imports.

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

Define the theme specific plugin.

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

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

Register the widget.

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

9.1.5 Form element plugin final steps

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

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

Afterwards, go to terminal and type the following command.

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

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

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

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

CHAPTER 10

Creating a new form handler plugin

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

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

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

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

10.1 Define and register the form handler plugin

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

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

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

10.1.1 path/to/sample_mail/fobi_form_handlers.py

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

Required imports.

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

Defining the Sample mail handler plugin.

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

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

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

Register the plugin

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

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

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

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

10.1.2 path/to/sample_mail/forms.py

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

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

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

Required imports.

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

Defining the form for Sample mail handler plugin.

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

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

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

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

10.1.3 Prioritise the execution order

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

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

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

10.1.4 Form handler plugin custom actions

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

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

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

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

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

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

10.1.5 Form handler plugin final steps

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

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

Afterwards, go to terminal and type the following command.

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

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

Creating a new form importer plugin

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

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

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

11.1 Define and register the form importer plugin

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

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

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

11.1.1 path/to/sample_importer/fobi_form_importers.py

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

Required imports.

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

Defining the Sample importer plugin.

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

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

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

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

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

    field_type_prop_name = 'field_type'
    position_prop_name = 'order'

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

(continues on next page)

(continued from previous page)

```
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 _
```

(continues on next page)

(continued from previous page)

```
# 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):
    """Sample importer wizard view."""

    form_list = [SampleImporterStep1Form, SampleImporterStep2Form]

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

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

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

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

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

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

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

(continues on next page)

(continued from previous page)

```
)

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).

CHAPTER 12

Creating a form callback

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

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

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

See the callback example below.

Required imports.

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

Define and register the callback

```
class SampleFooCallback(FormCallback):
    """Sample foo callback."""

    stage = CALLBACK_FORM_VALID

    def callback(self, form_entry, request, form):
        """Define your callback code here."""
        print("Great! Your form is valid!")

form_callback_registry.register(SampleFooCallback)
```

Add the callback module to `INSTALLED_APPS`.

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

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.

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'

```

(continues on next page)

(continued from previous page)

```

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',

```

(continues on next page)

(continued from previous page)

```

        '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

```

(continues on next page)

(continued from previous page)

```

— *** 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):
    """My simple theme, inherited from `SimpleTheme` theme."""

```

(continues on next page)

(continued from previous page)

```
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 %}
```

15.1 Basics

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

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

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

15.2 Bundled form wizard handler plugins

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

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

Integration with third-party apps and frameworks

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

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

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

- `IntegrationFormElementPlugin`
- `IntegrationFormHandlerPlugin`

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

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

16.1 Sample *IntegrationFormElementPlugin*

Sample is taken from [here](#).

16.1.1 base.py

Define the form element plugin.

```
from django.utils.translation import ugettext_lazy as _  
  
from rest_framework.fields import EmailField  
  
from fobi.base import IntegrationFormFieldPlugin
```

(continues on next page)

(continued from previous page)

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

class EmailInputPlugin(IntegrationFormFieldPlugin,
                       DRFSubmitPluginFormDataMixin):
    """EmailField plugin."""

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

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

16.1.2 fobi_integration_form_elements.py

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

```
from fobi.base import integration_form_element_plugin_registry
from .base import EmailInputPlugin

integration_form_element_plugin_registry.register(EmailInputPlugin)
```

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

16.2 Sample *IntegrationFormHandlerPlugin*

Sample is taken from [here](#).

16.2.1 base.py

Define the form handler plugin.

```
import logging
from mimetypes import guess_type
import os

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

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

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

from . import UID


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

    Can be used only once per form.
    """

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

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

        serializer = kwargs['serializer']

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

        rendered_data = form_handler_plugin.get_rendered_data(
            serializer.validated_data,
            field_name_to_label_map,
            base_url
        )

        files = self._prepare_files(request, serializer)

        form_handler_plugin.send_email(rendered_data, files)

    def _prepare_files(self, request, serializer):
```

(continues on next page)

(continued from previous page)

```
"""Prepares the files for being attached to the mail message."""
files = {}

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

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

    return files
```

16.2.2 fobi_integration_form_handlers.py

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

```
from fobi.base import integration_form_handler_plugin_registry
from .base import MailHandlerPlugin

integration_form_handler_plugin_registry.register(MailHandlerPlugin)
```

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

CHAPTER 17

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.

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.

20.1 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.

20.1.1 Fields

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

- Boolean (checkbox)
- Date
- DateTime
- Date drop down (year, month, day selection drop-downs)
- Decimal
- Duration
- Email*
- File
- Float
- Hidden*
- Input
- IP address*

- Integer
- Null boolean
- Password*
- Radio select (radio button)
- Range select
- Select (drop-down)
- Select model object (drop-down)
- Select multiple (drop-down)
- Select multiple model objects (drop-down)
- Slider
- Slug*
- Text*
- Textarea*
- Time
- URL*

20.1.2 Content/presentation

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

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

20.1.3 Security

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

20.1.4 MPTT fields

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

20.1.5 Test

Test plugins are made for dev purposes only.

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

20.2 Bundled form handler plugins

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

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

20.3 Bundled themes

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

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

Third-party plugins and themes

List of remarkable third-party plugins:

- [fobi-phonenum](#) - A Fobi PhoneNumber form field plugin. Makes use of the *phonenum-ber_field.formfields.PhoneNumberField* and *phonenum-ber_field.widgets.PhoneNumberPrefixWidget*.

The following HTML5 fields are supported in corresponding bundled plugins:

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

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

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

Loading initial data using GET arguments

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

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

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

Dynamic initial values

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

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

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

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

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

Note, that you should not provide the `fobi_dynamic_values`. as a prefix. Currently, the following variables are available

in the *fobi.context_processors.dynamic_values* context processor:

```
- request: Stripped HttpRequest object.

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

    - request.get_full_path(): Returns the path, plus an appended query
      string, if applicable.

    - request.is_secure(): Returns True if the request is secure; that
      is, if it was made with HTTPS.

    - request.is_ajax(): Returns True if the request was made via an
      XMLHttpRequest, by checking the HTTP_X_REQUESTED_WITH header for the
      string 'XMLHttpRequest'.

    - request.META: A stripped down standard Python dictionary containing
      the available HTTP headers.

        - HTTP_ACCEPT_ENCODING: Acceptable encodings for the response.

        - HTTP_ACCEPT_LANGUAGE: Acceptable languages for the response.

        - HTTP_HOST: The HTTP Host header sent by the client.

        - HTTP_REFERER: The referring page, if any.

        - HTTP_USER_AGENT: The client's user-agent string.

        - QUERY_STRING: The query string, as a single (un-parsed) string.

        - REMOTE_ADDR: The IP address of the client.

    - request.user: Authenticated user.

        - request.user.email:

        - request.user.get_username(): Returns the username for the user.
          Since the User model can be swapped out, you should use this
          method instead of referencing the username attribute directly.

        - request.user.get_full_name(): Returns the first_name plus the
          last_name, with a space in between.

        - request.user.get_short_name(): Returns the first_name.

        - request.user.is_anonymous():

- now: datetime.datetime.now()

- today: datetime.date.today()
```

Submitted form element plugins values

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

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

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

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

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

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

Rendering forms using third-party libraries

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

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

26.1 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 %}
```

26.2 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 %}
```

(continues on next page)

(continued from previous page)

```
{% endblock form_non_field_and_hidden_errors %}  
  
{% form form %}
```

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

Import/export forms

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

28.1 Available translations

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

- Dutch
- German
- Russian
- French

28.2 Overriding translations

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

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

- custom settings
- custom locales directory

Run the example project as follows:

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

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

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

```
FOBI_SORT_PLUGINS_BY_VALUE = True
```

CHAPTER 29

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`.

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

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

To test with all supported Python/Django versions type:

```
tox
```

To test against specific environment, type:

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

To test just your working environment type:

```
./runtests.py
```

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

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

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

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

30.1 Browser tests

For browser tests you may choose between Firefox, headless Firefox and PhantomJS. PhantomJS is faster, headless Firefox is fast as well, but normal Firefox tests tell you more (as you see what exactly happens on the screen). Both

cases require some effort and both have disadvantages regarding the installation (although once you have them installed they work perfect).

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

For PhantomJS you need to have NodeJS installed.

30.1.1 Set up ChromeDriver

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

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

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

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

After that your Selenium tests would work.

30.1.2 Set up Firefox 47

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

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

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

After that your Selenium tests would work.

30.1.3 Set up headless Firefox

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

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

2. Run the tests using headless Firefox.

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

Or run tox tests using headless Firefox.

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

Or run specific tox tests using headless Firefox.

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


30.1.4 Setup PhantomJS

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

1. Install PhantomJS and dependencies.

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

2. Specify the PHANTOM_JS_EXECUTABLE_PATH setting. Example:

```
PHANTOM_JS_EXECUTABLE_PATH = ""
```

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

Writing documentation

Keep the following hierarchy.

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


CHAPTER 32

Troubleshooting

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

CHAPTER 33

Contributing

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

CHAPTER 34

License

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

CHAPTER 35

Support

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

CHAPTER 36

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

CHAPTER 37

Screenshots

37.1 Bootstrap3 theme

37.1.1 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](#)

37.1.2 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

37.1.3 View/edit form

37.1.3.1 Form elements

Build your forms

Edit

View

Form My test form was created successfully.

Edit form

Elements

Handlers

Properties

Add elements to your form

Fields marked with * are required

Choose form element to add ▾

Save ordering

Build your forms

Edit

View

Edit form

Elements

Handlers

Properties

Add elements to your form

Fields marked with * are required

Choose form element to add ▾

Name (Text)*

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

Description (Textarea)

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

Content

Content video

Content text

Content image

Fields

Select multiple

Textarea

Select model object

URL

Text

DateTime

Radio

File

Boolean

Date

Integer

Hidden

Password

Email

Select

Testing

Dummy

ordering

© django-fobi example site 2014

Build your forms

Add "URL" element to the form

Fields marked with * are required

Label*

Name*

Help text

Initial

Max length*

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

37.1.3.2 Form handlers

Build your forms

Edit

View

The form element plugin "URL" was added successfully.

Edit form

Elements

Handlers

Properties

Add handlers to your form

Choose form handler to add ▾

© django-fobi example site 2014

Build your forms Edit View

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

Edit form

Elements Handlers Properties

Add handlers to your form

| Handler | Actions |
|----------|---|
| DB store | ✕ Delete ☰ View entries |

Choose form handler to add ▾

- DB store
- Mail
- HTTP Repost

© django-fobi example site 2014

Build your forms

Add "Mail" handler to the form

Fields marked with * are required

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

© django-fobi example site 2014

[Build your forms](#)**Edit**[View](#)

The form handler plugin "Mail" was added successfully.

Edit form

[Elements](#)**Handlers**[Properties](#)

Add handlers to your form

[Choose form handler to add ▼](#)**Handler****Actions**

DB store

[✕ Delete](#) [☰ View entries](#)Mail [✎ Edit](#) [✕ Delete](#)

Build your forms

Edit

View

Edit form

Elements

Handlers

Properties

Form properties

Fields marked with * are required

Name*

My test form

☒ Public? ?

Success page title

Thank you!

Custom message title to display after valid form is submitted

Success page body

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

Custom message text to display after valid form is submitted

Submit changes

Build your forms

Edit

View

View form

Fields marked with * are required

Name*

n.n.

Description

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

Website

http://delusionalinsanity.com/portfolio/

Submit

© django-fobi example site 2014

Build your forms

Form My test form was submitted successfully.

Thank you!

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

© django-fobi example site 2014

Build your forms

Add "Content video" element to the form

Fields marked with * are required

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

Add

© django-fobi example site 2014

Build your forms

Add "Boolean" element to the form

Fields marked with * are required

Label* 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 ☐[Add](#)

© django-fobi example site 2014

Build your forms Edit View

Elements ordering edited successfully.

Edit form

Elements Handlers Properties

Add elements to your form

Fields marked with * are required

Choose form element to add ▾

(Content video)



Edit Delete

Name (Text)*

Edit Delete

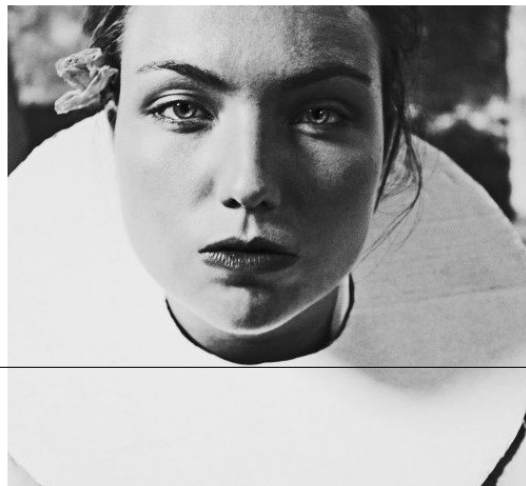
Description (Textarea)

Edit Delete

Website (URL)

Edit Delete

(Content image)



Edit Delete

Build your forms Edit View

View form

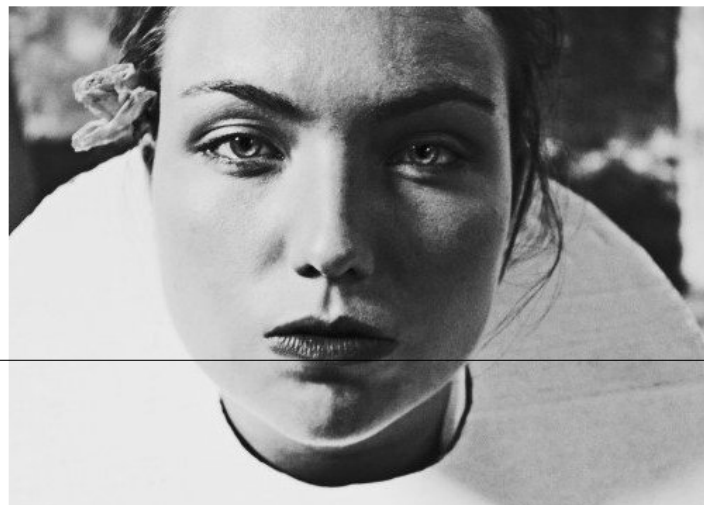
Fields marked with * are required



Name*

Description

Website



37.2 Simple theme

37.2.1 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):


37.2. Simple theme

115

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

Home > Fobi > My test form

Edit form

Elements
Handlers
Properties

Change form elements

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

(Content video):

Delusional Insanity - To Far Beyond

Edit Delete

Name (Text):

Edit Delete

Description (Textarea):

Edit Delete

Website (URL):

Edit Delete

(Content image):

116

Chapter 37. Screenshots

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: | <input type="text"/> |
| Name: | <input type="text"/> |
| Initial: | <input type="text"/> |
| Max length: | <input type="text" value="255"/> |
| Required: | <input type="checkbox"/> |

Add

Django administration

Welcome, **test_admin**. Log out

[Home](#) > [Fobi](#) > [My test form](#)

Edit form

Elements

Handlers

Properties

Change form handlers

Add form handler +

| Handler | Actions |
|----------|-------------|
| DB store | Delete |
| Mail ? | Edit Delete |

Edit form

Elements Handlers **Properties**

Change form properties

Name:

Public?



Makes your form visible to the public.

Success page
title:

Custom message title to display after valid form is submitted

Success page
body:

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

Custom message text to display after valid form is submitted

Save

View form



Name:

Description:

Website:



Contents:

38.1 Quick start

Tutorial for very quick start with `django-fobi`. Consists of several parts listed below:

- Part 1: Standard Django installation
- Part 2: Integration **with** DjangoCMS (coming soon)

38.1.1 Part 1: standard Django installation

Example project code available [here](#).

38.1.1.1 Installation and configuration

38.1.1.1.1 Install the package in your environment.

```
pip install django-fobi
```

38.1.1.1.2 INSTALLED_APPS

Add `fobi` core and the plugins to the `INSTALLED_APPS` of the your `settings` module.

1. The core.

```
'fobi',
```

2. The preferred theme. Bootstrap 3 theme is the default. If you have chosen a different theme, update the value of `FOBI_DEFAULT_THEME` accordingly.

```
'fobi.contrib.themes.bootstrap3',
```

3. The form field plugins. Plugins are like blocks. You are recommended to have them all installed. Note, that the following plugins do not have additional dependencies, while some others (like `fobi.contrib.plugins.form_elements.security.captcha` or `fobi.contrib.plugins.form_elements.security.recaptcha` would require additional packages to be installed. If so, make sure to have installed and configured those dependencies prior adding the dependant add-ons to the `settings` module.

```
'fobi.contrib.plugins.form_elements.fields.boolean',
'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple',
'fobi.contrib.plugins.form_elements.fields.date',
'fobi.contrib.plugins.form_elements.fields.date_drop_down',
'fobi.contrib.plugins.form_elements.fields.datetime',
'fobi.contrib.plugins.form_elements.fields.decimal',
'fobi.contrib.plugins.form_elements.fields.email',
'fobi.contrib.plugins.form_elements.fields.file',
'fobi.contrib.plugins.form_elements.fields.float',
'fobi.contrib.plugins.form_elements.fields.hidden',
'fobi.contrib.plugins.form_elements.fields.input',
'fobi.contrib.plugins.form_elements.fields.integer',
'fobi.contrib.plugins.form_elements.fields.ip_address',
'fobi.contrib.plugins.form_elements.fields.null_boolean',
'fobi.contrib.plugins.form_elements.fields.password',
'fobi.contrib.plugins.form_elements.fields.radio',
'fobi.contrib.plugins.form_elements.fields.regex',
'fobi.contrib.plugins.form_elements.fields.select',
'fobi.contrib.plugins.form_elements.fields.select_model_object',
'fobi.contrib.plugins.form_elements.fields.select_multiple',
'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects',
'fobi.contrib.plugins.form_elements.fields.slug',
'fobi.contrib.plugins.form_elements.fields.text',
'fobi.contrib.plugins.form_elements.fields.textarea',
'fobi.contrib.plugins.form_elements.fields.time',
'fobi.contrib.plugins.form_elements.fields.url',
```

4. The presentational form elements (images, texts, videos).

```
'easy_thumbnails', # Required by `content_image` plugin
'fobi.contrib.plugins.form_elements.content.content_image',
'fobi.contrib.plugins.form_elements.content.content_image_url',
'fobi.contrib.plugins.form_elements.content.content_text',
'fobi.contrib.plugins.form_elements.content.content_video',
```

5. Form handlers. Note, that some of them may require database sync/migration.

```
'fobi.contrib.plugins.form_handlers.db_store',
'fobi.contrib.plugins.form_handlers.http_repost',
'fobi.contrib.plugins.form_handlers.mail',
'fobi.contrib.plugins.form_handlers.mail_sender',
```

Putting all together, you would have something like this.

```
INSTALLED_APPS = (
    # Used by fobi
```

(continues on next page)

(continued from previous page)

```

'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_image_url',
'fobi.contrib.plugins.form_elements.content.content_text',
'fobi.contrib.plugins.form_elements.content.content_video',

# Form handlers
'fobi.contrib.plugins.form_handlers.db_store',
'fobi.contrib.plugins.form_handlers.http_repost',
'fobi.contrib.plugins.form_handlers.mail',
'fobi.contrib.plugins.form_handlers.mail_sender',

# ...

```

(continues on next page)

(continued from previous page)

```
)
```

38.1.1.1.3 TEMPLATE_CONTEXT_PROCESSORS

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

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join('path', 'to', 'your', 'templates')],
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                'fobi.context_processors.theme', # Important!
                'fobi.context_processors.dynamic_values', # Optional
            ],
            'loaders': [
                'django.template.loaders.filesystem.Loader',
                'django.template.loaders.app_directories.Loader',
                'admin_tools.template_loaders.Loader',
            ],
            'debug': DEBUG_TEMPLATE,
        },
    ],
]
```

38.1.1.1.4 urlpatterns

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

```
urlpatterns = [
    # ...

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

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

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

    # ...
]
```

38.1.1.1.5 Update the database

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

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

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

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

38.1.1.1.6 Specify the active theme

Specify the default theme in your `settings` module.

```
FOBI_DEFAULT_THEME = 'bootstrap3'
```

38.1.1.1.7 Permissions

`fobi` has been built with permissions in mind. Every single form element plugin or handler is permission based. If user hasn't been given permission to work with a form element or a form handler plugin, he won't be. If you want to switch the permission checks off, set the value of `FOBI_RESTRICT_PLUGIN_ACCESS` to `False` in your *settings* module.

```
FOBI_RESTRICT_PLUGIN_ACCESS = False
```

Otherwise, after having completed all the steps above, do log into the Django administration and assign the permissions (to certain user or a group) for every single form element or form handler plugin. Bulk assignments work as well.

- <http://yourdomain.com/admin/fobi/formelement/>
- <http://yourdomain.com/admin/fobi/formhandler/>

Also, make sure to have the Django model permissions set for following models:

- `fobi.models.FormEntry`
- `fobi.models.FormElementEntry`
- `fobi.models.FormHandlerEntry`
- `fobi.contrib.plugins.form_handlers.db_store.models.SavedFormDataEntry`

38.1.2 Part 2: Integration with DjangoCMS

Coming soon...

38.2 Release history and notes

Sequence based identifiers are used for versioning (schema follows below):

```
major.minor[.revision]
```

- It's always safe to upgrade within the same minor version (for example, from 0.3 to 0.3.4).
- Minor version changes might be backwards incompatible. Read the release notes carefully before upgrading (for example, when upgrading from 0.3.4 to 0.4).
- All backwards incompatible changes are mentioned in this document.

38.2.1 0.17.1

2021-01-25

Note: Release dedicated to defenders of Armenia and Artsakh (Nagorno Karabakh) and all the victims of Turkish and Azerbaijani aggression.

- Replace outdated *simplejson* with *json*.

38.2.2 0.17

2020-12-28

Note: Release dedicated to defenders of Armenia and Artsakh (Nagorno Karabakh) and all the victims of Turkish and Azerbaijani aggression.

- Deprecate support for the following Django versions: 1.11, 2.0 and 2.1.

38.2.3 0.16.4

2020-04-08

- Replace *admin_static* templatetag module with *static*, since the latter is available in Django 1.11, which is now the minimal supported version.

38.2.4 0.16.3

2020-04-04

- Fixes in *invisible_recaptcha* plugin. Treat empty `SITE_KEY` as error.

38.2.5 0.16.2

2020-04-03

- Conditional use of deprecated *admin_static* templatetag module.

38.2.6 0.16.1

2020-03-18

- Fix the `datetime` is not JSON serializable error.

38.2.7 0.16

2020-01-11

Note: This release drops support for Python 2.7 and 3.4 and sets the minimum requirement version for Django to 1.11. If you don't meet these requirements - do not upgrade.

- Add support for Django 3.0.
- Drop support Django 1.8, 1.9, 1.10. Minimal supported Django version is 1.11.
- Tested against Python 3.8.
- Drop Python 2.7 support.
- Drop Python 3.4 support.
- Tested with Django REST Framework 3.11.x. All older versions of Django REST Framework should still work, although they are no longer tested.
- Tested with FeinCMS 1.17.x. No code changes at all. All older versions of FeinCMS should still work, although they are no longer tested.

38.2.8 0.15

2019-06-20

Note: Although this release does not contain major backwards incompatible changes, some things you have had working in past might break.

- The following settings `FOBI_FAIL_ON_ERRORS_IN_FORM_ELEMENT_PLUGINS`, `FOBI_FAIL_ON_ERRORS_IN_FORM_HANDLER_PLUGINS`, `FOBI_FAIL_ON_MISSING_INTEGRATION_FORM_ELEMENT_PLUGINS` and `FOBI_FAIL_ON_MISSING_INTEGRATION_FORM_HANDLER_PLUGINS` used to be set to `False` by default (silencing exceptions). It's no longer so. Exceptions will cause errors from now on (you might get error 500). If you want old behaviour, set them to `False`.

38.2.9 0.14

2019-05-24

- Tested again Django 2.2 and Python 3.7.
- Use `ChromeDriver` (for browser tests).
- Wiping out (as stated many versions ago) support for Django < 1.8.
- Reactivate travis (although not all tests yet pass, while they do pass locally; to be solved in future).

38.2.10 0.13.10

2019-02-28

- Add `mail_sender` DRF integration plugin.

38.2.11 0.13.9

2019-02-28

Note: Release supported by [Goldmund](#), [Wyldebeast](#) & [Wunderliebe](#).

- Add *mail_sender* form handler plugin.
- Upgrade test suite.

38.2.12 0.13.8

2019-01-07

Note: Release supported by [Goldmund](#), [Wyldebeast](#) & [Wunderliebe](#).

- Make it easier to redirect to a new URL (on success) in integration apps.

38.2.13 0.13.7

2018-10-01

- Make fobi wagtail integration work with wagtail 2.

38.2.14 0.13.6

2018-08-16

- Completely wiping out `django-autoslug` in requirements (for now it has been replaced with `django-autoslug-iplweb`).

38.2.15 0.13.5

2018-08-12

- Minor fixes in code, tests and docs.

38.2.16 0.13.4

2018-08-11

- Django 2.1 support (initial).
- Upgrade test suite.

38.2.17 0.13.3

2018-07-10

- Minor fixes admin export of form entries (Postgres).

38.2.18 0.13.2

2018-03-12

- Choices are now required fields for *checkbox_select_multiple*, *radio*, *select*, *select_multiple* and *select_multiple_with_max* form elements.
- The *active_date_from*, *active_date_to*, *inactive_page_title*, *inactive_page_message* added to forms the import/export flow.

38.2.19 0.13.1

2018-02-28

- Minor fixes in recaptcha plugin.
- Add *max_length* to textarea plugin.

38.2.20 0.13

2018-02-25

Note: This release contains minor backwards incompatible changes. A number of new database fields have been added to the *FormEntry* model (*active_date_from*, *active_date_to*, *inactive_page_title*, *inactive_page_message*). You will need to migrate your database.

- Implement disabling forms based on dates. Note, that although the feature is landed into the core, contrib packages (such as Foundation 5 theme, Simple theme, Django CMS admin style theme, DRF integration app, FeinCMS integration app, Mezzanine integration app, DjangoCMS integration app and Wagtail app) are not yet updated to reflect these changes. It's a work in progress to be done in the nearest minor releases. Check the [issue #143](#) for state of the progress.

38.2.21 0.12.20

2018-02-24

- Minor Python 2 fixes.

38.2.22 0.12.19

2018-02-21

- Make it possible to sort on plugin name instead of `uid`.

38.2.23 0.12.18

2018-02-19

- Improved form element add drop-down order.
- Add initial migrations for DjangoCMS integration app.
- Add French translations.

- Implemented styles for `bleach` within `content_richtext` plugin.
- Documentation improvements.

38.2.24 0.12.17

2018-02-14

- Security fixes in `db_store` plugin.
- Minor fixes and cleanups.
- If CKEditor is installed, use it (rich text) for success page message.

38.2.25 0.12.16

2018-01-21

Note: Note, that this release contains minor backwards incompatible changes, that may slightly break your JS and/or styling.

The *form-horizontal* class attribute on the main form with elements (in templates) has been replaced with *fobi-form id* attribute.

Affected files:

- `src/fobi/templates/fobi/generic/add_form_element_entry_ajax.html`
- `src/fobi/templates/fobi/generic/edit_form_element_entry_ajax.html`
- `src/fobi/templates/fobi/generic/snippets/form_ajax.html`

If you have modified any of these, you might want to check and update your code.

- Invisible reCAPTCHA form plugin (security) added.
- Clean up templates.
- Introduce a `form_id` block. The following templates were affected.

38.2.26 0.12.15

2018-01-17

- Minor clean up of HTML templates (in some templates `hidden_form_element_wrapper_html_class` was used while others `form_element_wrapper_hidden_html_class`). Now `form_element_wrapper_hidden_html_class` is used everywhere. Some HTML files have been re-indented (2 spaces).
- Minor clean up of JS.
- Optionally limit file extensions for file upload element

38.2.27 0.12.14

2018-01-10

- Minor clean up of HTML templates (unused classes removed).
- Nicer captcha (simple captcha) for bootstrap3 and foundation5 themes.

38.2.28 0.12.13

2018-01-09

Note: Note, that this release contains minor backwards incompatible changes, that may slightly break your JS and/or styling.

The *form-horizontal* class attribute on the main form with elements (in templates) has been replaced with *fobi-form* id attribute.

Affected files:

- /src/fobi/contrib/themes/djangocms_admin_style_theme/static/djangocms_admin_style_theme/js/fobi.djangocms_admin_style_th
- /src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html
- /src/fobi/contrib/themes/foundation5/static/foundation5/js/foundation5_fobi_extras.js
- /src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html
- /src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_wizard_entry_ajax.html
- /src/fobi/contrib/themes/simple/static/simple/js/fobi.simple.edit.js
- /src/fobi/contrib/themes/simple/templates/simple/edit_form_entry_ajax.html
- /src/fobi/contrib/themes/simple/templates/simple/edit_form_wizard_entry_ajax.html
- /src/fobi/contrib/themes/bootstrap3/static/bootstrap3/js/bootstrap3_fobi_extras.js
- /src/fobi/templates/fobi/generic/edit_form_entry_ajax.html

If you have modified any of these, please double check that your edit views work and look properly.

- Fixes in simple and django-cms-admin-style themes (assets being loaded incorrectly since 0.12.6).
- Base template, as well as templates of all the themes have been made a little bit more flexible.
- Minor documentation fixes.

38.2.29 0.12.12

2018-01-03

- More consistent loading of assets (JS, CSS) in add- and edit-form-element-entry templates.
- Copyright and licenses year update.
- Minor fixes in the widgets.
- Clean up.

38.2.30 0.12.11

2017-12-27

- Clean up Django < 1.8 code.
- Clean up old code.

38.2.31 0.12.10

2017-12-24

- Django 2.0 support (experimental).
- (Temporary) replace `django-autoslug` package with `django-autoslug-iplweb`, which works fine for Django versions `>=1.8,<=2.0`.

38.2.32 0.12.9

2017-12-21

- Added content markdown plugin.

38.2.33 0.12.8

2017-12-19

- Added common form callback `AutoFormMail` for auto mailing the form data (without need to add the mail plugin to the form).

38.2.34 0.12.7

2017-12-17

- Minor fixes (in the add form element entry bootstrap3 template).
- Add content rich text plugin (based on CKEditor).
- Added common form callback `AutoFormDbStore` for auto saving the form data (without need to add the `db_store` plugin to the form).

38.2.35 0.12.6

2017-12-12

- Clean up.
- Minor documentation fixes.
- Minor plugin clean-up/fixes (`captcha`, `recaptcha`, `content_text`).
- Minor Python 3 fixes.
- Minor fixes in `FormWizard` `get_absolute_url` method.

38.2.36 0.12.5

2017-09-27

- Documentation fixes.

38.2.37 0.12.4

2017-09-27

- Documentation fixes.

38.2.38 0.12.3

2017-09-27

- Minor fixes.
- Reflect form-wizards support changes in the *foundation5* theme.
- Documentation fixes.

38.2.39 0.12.2

2017-08-02

- Some work on full form-wizards support in the *foundation5* theme.
- Django 1.11 fixes for customised widgets.
- Update example project requirements.

38.2.40 0.12.1

2017-07-31

- Reflect form-wizards support changes in the *simple* theme.
- Fixes in docs.

38.2.41 0.12

2017-06-28

- Wagtail integration (yet experimental).

38.2.42 0.11.13

2017-06-10

- Updated docs for DjangoCMS, FeinCMS and Mezzanine.
- Updated outdated Dutch translations.
- Improvements of the DRF integration app documentation.

- Duration field added.
- Support `DurationField` in `drf_integration` app.
- Minor fixes in docs.

38.2.43 0.11.12

2017-05-31

- Added a lot of field metadata to the `OPTIONS` call of `drf_integration` app.
- Appended a lot of sub-module `README` files to the main documentation.

38.2.44 0.11.11

2017-05-29

- Minor fixes in `content_text` `drf_integration` plugin.
- Added `imageurl` support to the `mailchimp_importer` plugin.

38.2.45 0.11.10

2017-05-26

- Minor fixes in form-wizards on Django 1.11.

38.2.46 0.11.9

2017-05-24

- Mezzanine integration updated to work with Mezzanine 4.2.3.
- Fixes in date-drop-down plugin when using form wizards.

38.2.47 0.11.8

2017-05-17

- `ContentImageURL` plugin added.
- Minor Python3 fixes in `db_store` plugin (related to export of forms to `xls` format).

38.2.48 0.11.7

2017-05-16

- Fixed in `fobi.integration` package related to Django 1.10 and higher.
- FeinCMS integration updated (only migrations added) to work with FeinCMS 1.13.2.
- DjangoCMS integration updated to work with DjangoCMS 3.4.3.

38.2.49 0.11.6

2017-05-15

- Minor fixes in `drf_integration` app, added GET/detail actions tests.

38.2.50 0.11.5

2017-05-15

- Added `date_drop_down` to `drf_integration` app.
- Fixed dependencies issue.
- Added dedicated requirements for specific Django versions.

38.2.51 0.11.4

2017-05-12

- Minor fixes in integration callbacks of the `drf_integration` sub-package.
- Added support for `content_image`, `content_text` and `content_video` plugins.
- Fixes in installable demo.

38.2.52 0.11.3

2017-05-10

- Concept of integration callbacks introduced and implemented for the `drf_integration` sub-package.

38.2.53 0.11.2

2017-05-09

- Minor fixes in `drf_integration` app.

38.2.54 0.11.1

2017-05-08

- Minor fixes in `decimal` plugin.
- Minor documentation improvements.

38.2.55 0.11

2017-05-07

This release is dedicated to my beloved [wife](#) for all the love and support she gave me through years. If you are a company looking for (female) developers in Groningen area (the Netherlands), do not hesitate to [contact her](#).

- Django REST framework integration. Check the Heroku demo [here](#).
- Documentation fixes.

- PEP8 code fixes.
- Minor setup fixes related to moved screen-shots file.
- Added helper scripts to test with Firefox in headless mode. Describe testing with Firefox in headless mode in documentation.
- Validate the `decimal` field plugin - quantize the decimal value to the configured precision.
- Minor fixes in the `float` field plugin.
- Minor improvements in complex form element plugins (`select`, `file`) and form handler plugins (`db_store`, `mail`, `http_repost`) in order to simplify integration plugins and reduce code duplication.
- Minor Python3 fixes in `range_select` and `slider` form element plugins.
- Minor Python3 fixes in `http_repost` and `mail` form handler plugins.

38.2.56 0.10.7

2017-03-13

- Several Django deprecation/moves fixes for better future compatibility.

38.2.57 0.10.6

2017-02-14

- Minor Python 3 fixes for integer, float and decimal fields.

38.2.58 0.10.5

2017-02-13

- Tested against Python 3.6.
- Initial (experimental) Django 1.11 support.

38.2.59 0.10.4

2017-01-11

- Minor fixes in Django admin.
- Various pep8 fixes.
- Fixes additions and improvements in/of docs.
- Add options to test with PhantomJS instead of Firefox.

38.2.60 0.10.3

2016-11-24

- Minor fixes.

38.2.61 0.10.2

2016-11-24

- Minor fixes.

38.2.62 0.10.1

2016-11-17

- Fixed captcha and re-captcha issues in form wizards.

38.2.63 0.10

2016-11-16

Note: Note, that this release contains minor backwards incompatible changes, that may break your code. Two additional arguments have been added to the `submit_plugin_form_data` method of the form element plugins. If you have written custom form element plugins - update your code.

- Added `form_entry_elements` and `kwargs` to the `submit_plugin_form_data` method of the form element plugins. Make sure to update your custom plugins if you have written any.
- Added tests for mailchimp integration plugin.
- Moving all plugins to base submodules of the correspondent sub packages.
- Add missing whitespace to the `help_text` of the `title` field of `FormEntry` and `FormWizardEntry` models.
- Disable GoogleAnalytics while testing (guess what - this change speeds up selenium tests twice).
- Docs updated.
- Helper scripts updated.
- Multiple pep8 fixes.

38.2.64 0.9.17

2016-11-13

Note: Announcing dropping support of Python 2.6 and Django 1.7. As of 0.9.17 everything is still backwards compatible with Django 1.7, but in future versions it will be wiped out.

- Value validations for Integer and Text Fields.
- Hide previous button in form wizard template for bootstrap3 on first step.

38.2.65 0.9.16

2016-11-10

- Introduced form titles (shown in view templates).
- Improved navigation of the form wizards.

38.2.66 0.9.15

2016-11-07

- Minor fixes.

38.2.67 0.9.14

2016-11-07

- Minor fixes.

38.2.68 0.9.13

2016-11-05

Note: Announcing dropping support of Django 1.5 and 1.6. As of 0.9.13 everything is still backwards compatible with versions 1.5 and 1.6, but in future versions compatibility with these versions will be wiped out.

- Fix backwards compatibility of *slider* and *range_select* plugins with Django versions 1.5 and 1.6.

38.2.69 0.9.12

2016-11-02

- Better debugging.
- Upgrade example FeinCMS integration to work with 1.12.

38.2.70 0.9.11

2016-11-01

- Fixes.

38.2.71 0.9.10

2016-11-01

- Fixed issue with custom labels in the *slider* plugin.
- Made *slider* plugin compatible with Django <= 1.6.
- Fixes *get_absolute_url* methods on *FormEntry* and *FormWizardEntry* models. #48

38.2.72 0.9.9

2016-10-31

- Make it possible to add custom ticks to the *slider* plugin.

38.2.73 0.9.8

2016-10-27

- Support multiple sliders in one form.

38.2.74 0.9.7

2016-10-27

- Improvements in the generic integration processor. #47
- Improved form wizard interface and navigation.
- Fixed a broken test.
- Added import/export functionality for form wizards.

38.2.75 0.9.6

2016-10-25

- Fixed InvalidQuery exception raised when attempting to export entry from a ‘DB store’ handler. #44
- Fixed ProgrammingError raised when using the ‘Export data to CSV/XLS’ action. #45

38.2.76 0.9.5

2016-10-25

- Minor fixes in *slider* and *select_range* plugins.

38.2.77 0.9.4

2016-10-24

- Fix issue with *select_multiple*, *select_multiple_model_objects* and *select_multiple_mptt_model_objects* being invalidated on the last step of the form wizard.

38.2.78 0.9.3

2016-10-24

- Change to *NumberInput* widget for all number inputs.
- Fixed issue with *slider* plugin missing labels if *Show endpoints as* is set to *Labeled ticks*.
- Link to edit form entry added to edit form wizard entry view.

38.2.79 0.9.2

2016-10-24

- Minor fixes.

38.2.80 0.9.1

2016-10-24

- Minor fixes.

38.2.81 0.9

2016-10-24

Note: Note, that this release contain minor backwards incompatible changes, that may break your existing code (your data is left intact). If you have written custom form element plugins you should update your code!

- The **:method:‘get_form_field_instances‘** and **:method:‘_get_form_field_instances‘** of the *fobi.base.FormElementPlugin* both accept two new optional arguments: *form_entry* and *form_element_entries* as well as ***kwargs*. Make sure to update your custom plugins if you have written any.
- Minor fixes in the form wizards: forms in intermediate steps do receive updates from the *submit_plugin_form_data* of the plugins.
- Fixed issue in the *base_bulk_change_plugins* function on Django 1.10.

38.2.82 0.8.10

2016-10-22

- Minor CSS improvements of the *slider* plugin.
- Fixed broken readthedocs requirements.

38.2.83 0.8.9

2016-10-22

- Simplified debugging (never set *FOBI_DEBUG* to True in production!).
- Major *slider* plugin improvements.

38.2.84 0.8.8

2016-10-21

- Minor *slider* plugin improvements (JavaScript).

38.2.85 0.8.7

2016-10-21

- Fixed issue of plugin media not being collected in the form wizard.

38.2.86 0.8.6

2016-10-21

- Functional improvements of *slider* plugin.

38.2.87 0.8.5

2016-10-20

- Add *range_select* and *slider* form field plugins.
- Fixed custom CSS classes not appearing in the rendered HTML of the field plugin/widget.
- Fixed issue with undefined file storage for form wizards. From now on the *FileSystemStorage* storage is used for wizard uploads.
- Fixed too much of extreme data view/export security of the *db_store* plugin.
- Backwards compatibility fixes for Django < 1.7.

38.2.88 0.8.4

2016-10-19

- Fix broken export (to JSON) of form entries.
- Fix broken import (from JSON) of form entries.

38.2.89 0.8.3

2016-10-18

- Minor fixes.

38.2.90 0.8.2

2016-10-18

- Minor fixes.

38.2.91 0.8.1

2016-10-17

- Minor fixes.

38.2.92 0.8

2016-10-17

Release supported by [Lund University Cognitive Science](#).

- Adding form-wizards functionality. Note, that at the moment only *bootstrap3* theme was updated to fully support the form wizards. Although, all other themes would by default support form-wizard functionality, they may not look as nice as they should be (to be fixed in 0.8.x releases shortly).
- The *six* package requirements increased to ≥ 1.8 .
- Tests comply with pep8.
- Fixed recently broken drag-and-drop ordering of the form elements.
- Fixed typo for HTML id “tab-form-elemenets” -> “tab-form-elements”. You may need to update your custom CSS/JS/HTML accordingly. See the listing 0.8.a for the files affected.
- An additional property *form_view_form_entry_option_class* has been added to all the themes. Change your custom CSS/JS/HTML accordingly. See the listing 0.8.b for the files affected.
- Fixed drag-and-drop not working for ordering of form elements. #43
- Fixed issue with non-proper rendering of the form-importer templates.

Note: Although this release does not contain backwards incompatible changes, there have been several changes in GUI and some parts of the generic HTML and themes were updated. If you have custom themes implemented, you should likely make some minor updates to the HTML in order to reflect the latest GUI changes. The following templates have been affected:

38.2.92.1 New files

- `src/fobi/contrib/plugins/form_handlers/db_store/templates/db_store/view_saved_form_wizard_data_entries.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/add_form_wizard_handler_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/add_form_wizard_handler_entry_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/create_form_wizard_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/create_form_wizard_entry_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/edit_form_wizard_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/edit_form_wizard_entry_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/form_wizards_dashboard.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/snippets/form_wizard_ajax.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/snippets/form_wizard_properties_snippet.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/snippets/form_wizard_snippet.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/view_form_wizard_entry.html`
- `src/fobi/contrib/themes/bootstrap3/templates/bootstrap3/view_form_wizard_entry_ajax.html`
- `src/fobi/templates/fobi/generic/add_form_wizard_handler_entry.html`
- `src/fobi/templates/fobi/generic/add_form_wizard_handler_entry_ajax.html`
- `src/fobi/templates/fobi/generic/create_form_wizard_entry.html`

- src/fobi/templates/fobi/generic/create_form_wizard_entry_ajax.html
- src/fobi/templates/fobi/generic/edit_form_wizard_entry.html
- src/fobi/templates/fobi/generic/edit_form_wizard_entry_ajax.html
- src/fobi/templates/fobi/generic/form_wizard_entry_submitted.html
- src/fobi/templates/fobi/generic/form_wizard_entry_submitted_ajax.html
- src/fobi/templates/fobi/generic/form_wizards_dashboard.html
- src/fobi/templates/fobi/generic/snippets/form_wizard_ajax.html
- src/fobi/templates/fobi/generic/snippets/form_wizard_properties_snippet.html
- src/fobi/templates/fobi/generic/snippets/form_wizard_snippet.html
- src/fobi/templates/fobi/generic/snippets/form_wizard_view_ajax.html
- src/fobi/templates/fobi/generic/view_form_wizard_entry.html
- src/fobi/templates/fobi/generic/view_form_wizard_entry_ajax.html

38.2.92.2 Existing files

- src/fobi/contrib/plugins/form_importers/mailchimp_importer/templates/mailchimp_importer/1.html
- src/fobi/contrib/plugins/form_importers/mailchimp_importer/views.py
- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html
- src/fobi/templates/fobi/generic/edit_form_entry_ajax.html

38.2.92.3 Additional listings

38.2.92.3.1 Listing 0.8.a “tab-form-elemenets” -> “tab-form-elements”

- src/fobi/contrib/themes/djangocms_admin_style_theme/static/djangocms_admin_style_theme/css/fobi.djangocms_admin_style_t
- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html
- src/fobi/contrib/themes/simple/static/simple/css/fobi.simple.edit.css
- src/fobi/contrib/themes/simple/templates/simple/edit_form_entry_ajax.html
- src/fobi/templates/fobi/generic/edit_form_entry_ajax.html
- src/fobi/templates/fobi/generic/edit_form_wizard_entry_ajax.html

38.2.92.3.2 Listing 0.8.b *form_view_form_entry_option_class* property

- src/fobi/contrib/themes/bootstrap3/fobi_themes.py
- src/fobi/contrib/themes/djangocms_admin_style_theme/fobi_themes.py
- src/fobi/contrib/themes/foundation5/fobi_themes.py
- src/fobi/contrib/themes/simple/fobi_themes.py

- `src/fobi/templates/fobi/generic/edit_form_wizard_entry_ajax.html`

38.2.93 0.7.1

2016-09-22

Release sponsored by [Goldmund](#), [Wyldebeast](#) & [Wunderliebe](#).

- Code comply with pep8.
- Minor fixes in selenium tests.

38.2.94 0.7

2016-09-13

Release sponsored by [Goldmund](#), [Wyldebeast](#) & [Wunderliebe](#).

- Initial Django 1.10 support.
- `django-localeurl` has been replaced with `i18n_patterns` in sample project.
- Minor fixes.

38.2.95 0.6.10

2016-09-11

- Moved `plugin_uid` field choices from model level to form level for *FormHandler* and *FormHandlerEntry* models. #37

38.2.96 0.6.9

2016-09-08

- Moved `plugin_uid` field choices from model level to form level for *FormElement* and *FormElementEntry* models. #37
- Fixed element “name” field stripping underscores issue. #33

38.2.97 0.6.8

2016-09-06

- Fixed changing order of the *FormElement*, *FormElementEntry*, *FormHandler* and *FormHandlerEntry* models.

38.2.98 0.6.7

2016-08-30

- Minor fixes in `db_store` plugin (Django 1.6 compatibility issue).
- Added `__str__` methods to models.
- Restrict queryset to form element entries related to the form entry in position calculation.

38.2.99 0.6.6

2016-05-12

- Fixed broken dependencies in demos.
- Minor fixes.
- Adding a new *Select multiple with max* plugin, which restricts max number of choices allowed to be chosen.

38.2.100 0.6.5

2015-12-24

- Minor fixes.

38.2.101 0.6.4

2015-12-24

- Minor fixes.

38.2.102 0.6.3

2015-12-23

- Minor fixes.

38.2.103 0.6.2

2015-12-22

- Make it possible to render a list of forms using custom template tag (not only on the dashboard page).

38.2.104 0.6.1

2015-12-21

- Documentation fixes.

38.2.105 0.6

2015-12-18

- Form importers (and as a part of it - MailChimp integration, which would allow to import forms from MailChimp into *django-fobi* using a user-friendly wizard).
- Improved Django 1.9 support.

38.2.106 0.5.19

2015-12-15

- New style urls everywhere.

38.2.107 0.5.18

2015-12-08

- Minor improvements. Adding request to the *get_form_field_instances* method of the *FormElementPlugin*.

38.2.108 0.5.17

2015-10-22

- Increased *easy-thumbnails* requirement to ≥ 2.1 for Python 3, since it was causing installation errors.
- Increased the *django-nine* requirement to $\geq 0.1.6$, since it has better Python 3 support.

38.2.109 0.5.16

2015-10-05

- Fixes in migrations on Django 1.7.

38.2.110 0.5.15

2015-09-17

- Minor fixes.

38.2.111 0.5.14

2015-09-13

- Add create/update dates to the forms. Adding initial migrations.

38.2.112 0.5.13

2015-09-01

- Translation fixes.
- Updating Dutch and Russian translations.

38.2.113 0.5.12

2015-08-29

- Export/import forms into/from JSON.
- Minor UI improvements. Adding “Service” tab in the edit view which contains links to export form to JSON and delete form.

Note: Although this release does not contain backwards incompatible changes, there have been several changes in GUI and some parts of the generic HTML and themes were updated. If you have custom themes implemented, you should likely make some minor updates to the HTML in order to reflect the latest GUI changes. The following templates have been affected:

38.2.113.1 New files

- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/import_form_entry.html
- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/import_form_entry_ajax.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/import_form_entry.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/import_form_entry_ajax.html
- src/fobi/contrib/themes/simple/templates/simple/import_form_entry.html
- src/fobi/contrib/themes/simple/templates/simple/import_form_entry_ajax.html
- src/fobi/templates/fobi/generic/import_form_entry.html
- src/fobi/templates/fobi/generic/import_form_entry_ajax.html

38.2.113.2 Existing files

- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/dashboard.html
- src/fobi/contrib/themes/djangocms_admin_style_theme/templates/djangocms_admin_style_theme/edit_form_entry_ajax.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/dashboard.html
- src/fobi/contrib/themes/foundation5/templates/foundation5/edit_form_entry_ajax.html
- src/fobi/contrib/themes/simple/templates/simple/dashboard.html
- src/fobi/contrib/themes/simple/templates/simple/edit_form_entry_ajax.html
- src/fobi/templates/fobi/generic/dashboard.html
- src/fobi/templates/fobi/generic/edit_form_entry_ajax.html

38.2.114 0.5.11

2015-08-20

- Minor improvements of the dynamic values feature. Forbid usage of django template tags in initial values.

38.2.115 0.5.10

2015-08-18

- Minor improvements of the initial dynamic values feature.

38.2.116 0.5.9

2015-08-17

- Minor fixes in the initial dynamic values feature.

38.2.117 0.5.8

2015-08-16

- Made it possible to define dynamic initials for form fields. Example initial dynamic values in the form (like `{{ request.path }}`).
- Minor fixes/improvements.

38.2.118 0.5.7

2015-08-03

- Minor Python 3 improvements.

38.2.119 0.5.6

2015-07-31

- *django-mptt* support through *select_mptt_model_object* and *select_multiple_mptt_model_objects* plugins.
- Python 3 fixes.

38.2.120 0.5.5

2015-06-30

- Change the *action* field of the *FormEntry* into a URL field; check if action exists.
- *captcha*, *recaptcha* and *honeypot* plugins have been made required in the form.
- Fix: take default values provided in the *plugin_data_fields* of the plugin form into consideration.

38.2.121 0.5.4

2015-05-21

- Minor Django 1.8 fixes.
- Improved texts/translations.

38.2.122 0.5.3

2015-05-02

- Minor fixes in the *mail* form handler plugin.

38.2.123 0.5.2

2015-04-26

- Make it possible to provide multiple *to* email addresses in the *mail* form handler plugin.
- DateTime picker widget added for Foundation5 theme for *date* and *datetime* plugins.
- Added more tests (more plugins tested).

38.2.124 0.5.1

2015-04-21

- DateTime picker widget added for Bootstrap 3 theme for `date` and `datetime` plugins.

38.2.125 0.5

2015-04-06

Note: Note, that this release contains minor backwards incompatible changes. The changes may affect your existing forms and data. Read the notes below carefully.

- Fixed previously wrongly labeled (in *AppConf*) add-ons/plugins (*fobi.contrib.plugins.form_handlers.db_store*, *fobi.contrib.apps.feincms_integration*, *fobi.contrib.apps.djangocms_integration*, *fobi.contrib.apps.mezzanine_integration*). Due to the change, you would likely have to rename a couple of database tables and update references accordingly. No migrations to solve the issue are included at the moment.

38.2.126 0.4.36

2015-04-03

- Handle non-ASCII characters `content_text` form element.

38.2.127 0.4.35

2015-03-28

- Fixed the issue with *allow_multiple* working incorrectly for form handler plugins. Fix the *db_store* plugin as well.

38.2.128 0.4.34

2015-03-27

- Minor fixes in the *Checkbox select multiple* and *Radio* plugins.
- Minified tox tests.

38.2.129 0.4.33

2015-03-26

- `Checkbox select multiple` field added.
- Minor improvements (styling) in the Foundation 5 theme.
- Initial configuration for tox tests.
- Clean up requirements (for example setups and tests).

38.2.130 0.4.32

2015-03-25

- Updated missing parts in the Russian translations.
- Minor API improvements. From now on, the *run* method of form handlers may return a tuple (bool, mixed). In case of errors it might be (False, err).
- Minor code clean ups.

38.2.131 0.4.31

2015-03-23

- When path of the uploaded file (plugins) doesn't yet exist, create it, instead of failing.

38.2.132 0.4.30

2015-03-23

- From now on submitted files are sent as attachments in the mail plugin.
- Documentation improvements. Adding information of rendering forms using *django-crispy-forms* or alternatives.
- Minor fixes.

38.2.133 0.4.29

2015-03-20

- **Decimal** field added.
- **Float** field added.
- **Slug** field added.
- **IP address** field added.
- **Null boolean** field added.
- **Time** field added.
- From now on using *simplejson* package in favour of *json*, since it can handle decimal data.
- Minor improvements of the date plugins (*datetime*, *date*).

38.2.134 0.4.28

2015-03-13

- Fix improperly picked configurations of the *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects* plugin.
- Long identifiers of models can now be safely used in foreign key plugins (such as *fobi.contrib.plugins.form_elements.fields.select_model_object_plugin* and *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects* plugins).

- Fixed admin bulk change of the plugins.
- From now it's possible to have some control/configure the following plugins for the submitted value:
 - `fobi.contrib.plugins.form_elements.fields.select_model_object`
 - `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects`

The only thing needs to be done is to specify the appropriate variable in the settings module of the project (`settings.py`).

- `FOBI_FORM_ELEMENT_SELECT_MODEL_OBJECT_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MODEL_OBJECTS_SUBMIT_VALUE_AS`

Allowed values are: “val”, “repr”, “mix”.

38.2.135 0.4.27

2015-03-12

- Temporary allow the `fobi.contrib.plugins.form_handlers.db_store` plugin to be used multiple times per form, until the bug with not being able to assign the `db_store` plugin to the form due to incorrect handling of restrictions (`allow_multiple`) introduced in previous version is properly fixed.
- From now it's possible to have some control/configure the following plugins for the submitted value:
 - `fobi.contrib.plugins.form_elements.fields.radio`
 - `fobi.contrib.plugins.form_elements.fields.select`
 - `fobi.contrib.plugins.form_elements.fields.select_multiple`

The only thing needs to be done is to specify the appropriate variable in the settings module of the project (`settings.py`).

- `FOBI_FORM_ELEMENT_RADIO_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_SUBMIT_VALUE_AS`
- `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_SUBMIT_VALUE_AS`

Allowed values are: “val”, “repr”, “mix”.

38.2.136 0.4.26

2015-03-06

- Validate `fobi.contrib.plugins.form_elements.fields.email`, `fobi.contrib.plugins.form_elements.fields.integer` and `fobi.contrib.plugins.form_elements.fields.url` plugins initial values.
- Properly show field types “checkbox” and “radio” in the `input` plugin (as previously they showed up too large).
- It's now possible to restrict multiple usage of form handler plugins via `allow_multiple` property. In case if it's set to `False`, the plugin can be used once only (per form). Default value is `True`. The `db_store` plugin is from now on allowed to be used only once (per form).

38.2.137 0.4.25

2015-03-04

- Post-fix in the discover module (moved logging definition up).

38.2.138 0.4.24

2015-03-04

- The management command `migrate_03_to_04` intended to migrate 0.3.x branch data to > 0.4.x branch data, has been renamed to `fobi_migrate_03_to_04`.
- Add missing app config declaration for the `db_store` form handler plugin.
- Add missing app config for the core `fobi` package.
- Improved autodiscover for Django>=1.7. Fix exception when using a dotted path to an `AppConfig` in `INSTALLED_APPS` (instead of using the path to the app: ex. “`path.to.app.apps.AppConfig`” instead of “`path.to.app`”).

38.2.139 0.4.23

2015-03-04

- Fix improper initial value validation for Select-like (`radio`, `select` and `select_multiple`) plugins.

38.2.140 0.4.22

2015-03-03

- Fix replace system-specific path separator by a slash on file urls.
- Fix empty options appearing in the Select-like (`radio`, `select` and `select_multiple`) plugins and unified the processing of the raw choices data.
- Validate the initial value for Select-like (`radio`, `select` and `select_multiple`) plugins.

38.2.141 0.4.21

2015-02-28

- The `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects` plugin added.

38.2.142 0.4.20

2015-02-27

- Make it possible to load initial form data from GET variables.
- Remove “button” and “submit” types from `fobi.contrib.elements.fields.input` form element plugin.
- The `fobi.contrib.plugins.form_elements.fields.select_model_object` plugin no longer returns an absolute URL to the admin of the chosen model object instance. Instead, it contains the app label, model name, pk and the repr of it separated by dots. Example: “`comments.comment.5.Lorem ipsum`”.
- Minor fixes in from `fobi.contrib.elements.fields.file` plugin. Use system path separator on from `fobi.contrib.elements.fields.file.FILES_UPLOAD_DIR` default setting.
- Minor documentation improvements.

38.2.143 0.4.19

2015-02-15

- Some more work on future (Django 1.9) compatibility.
- Replace bits of backwards-/forwards- compatibility code with equal code parts of `django-nine`.

38.2.144 0.4.18

2015-02-13

- From now on it's possible to localise (translated) URLs.
- Safe way to get the user model for Django `>= 1.5.* <= 1.8.*`

38.2.145 0.4.17

2015-02-12

- Fix circular imports by proper referencing of the user model in foreign key and many-to-many relations (`settings.AUTH_USER_MODEL` instead of `django.contrib.auth.get_user_model`).
- Minor documentation improvements.

38.2.146 0.4.16

2015-02-10

- A new theme `djangoCMS_admin_style_theme` added.
- Making `fobi.fields.NoneField` always valid.
- Initial work on Django 1.8 and 1.9 support.
- Minor fixes.

38.2.147 0.4.15

2015-01-27

- Fix the “large” checkboxes in edit mode - now shown small as they should be.

38.2.148 0.4.14

2015-01-26

- German translations added.

38.2.149 0.4.13

2015-01-15

- Remove an *ipdb* statement from base integration processor *fobi.integration.processors.IntegrationProcessor*.
- Added information in the docs about FeinCMS demo part on heroku demo.
- Make sure values of form elements declared not to have a value (*has_value* property is set to *False*) aren't being saved in the *db_store* plugin.
- Remove redundant static assets (package size decreased).

38.2.150 0.4.12

2015-01-14

- Fix empty options appearing in the Select-like plugins and unified the processing of the raw choices data.
- Update the *vishap* package requirement to latest stable 0.1.3.
- Support for wheel packages.

38.2.151 0.4.11

2014-12-29

- Styling fixes in the *radio* button field of the *bootstrap3* theme.
- Fixed *db_store* issue with CSV/XLS export failing on Django 1.7.

38.2.152 0.4.10

2014-12-28

- Minor fixes in FeinCMS integration app.

38.2.153 0.4.9

2014-12-28

- Third party app integration (at the moment, FeinCMS, DjangoCMS, Mezzanine) had been generalised and unified.
- Mention the Heroku live demo in the docs.
- Minor CSS fixes in the *simple* theme.

38.2.154 0.4.8

2014-12-25

- More verbose debugging.

38.2.155 0.4.7

2014-12-24

- Temporary left out the “cloneable” column from the dashboard templates.
- Fixed broken imports in CAPTCHA plugin.
- Fixed broken imports in ReCAPTCHA plugin.

38.2.156 0.4.6

2014-12-23

- Updated requirements for the `vishap` package to avoid the `six` version conflicts.
- Minor documentation fixes.

38.2.157 0.4.5

2014-12-17

- ReCAPTCHA field added.
- Mezzanine integration app added.
- Remove redundant dependencies (`django-tinymce`).
- Minor improvements of the discover module.

38.2.158 0.4.4

2014-12-06

- Documentation improvements.
- Updated Dutch and Russian translations.
- Minor fixes related to lazy translations.

38.2.159 0.4.3

2014-12-05

- Make sure values of form elements declared not to have a value (`has_value` property is set to `False`) aren't being saved in the `db_store` plugin.
- Apply that to the `honeypot` and `captcha` plugins.

38.2.160 0.4.2

2014-12-04

- Helper script (management command) in order to migrate `django-fobi==0.3.*` data to `django-fobi==0.4.*` data (caused by renaming the `birthday` field to `date_drop_down` - see the release notes of 0.4 below). Follow the steps precisely in order to painlessly upgrade your `django-fobi==0.3.*` to `django-fobi==0.4.*`:
 1. Install `django-fobi>=0.4.2`:

```
pip install django-fobi>=0.4.2
```

2. In your settings change the:

```
'fobi.contrib.plugins.form_elements.fields.birthday'
```

to:

```
'fobi.contrib.plugins.form_elements.fields.date_drop_down'
```

3. Run the `migrate_03_to_04` management command. Note, that as of version 0.4.24, the `migrate_03_to_04` command has been renamed to `fobi_migrate_03_to_04`:

```
./manage.py migrate_03_to_04
```

38.2.161 0.4.1

2014-12-04

- Fixes in Foundation5 and Simple themes related to the changes in error validation/handling of hidden fields.

38.2.162 0.4

2014-12-03

Note: Note, that this release contains minor backwards incompatible changes. The changes may affect your existing forms and data. Read the notes below carefully (UPDATE 2014-12-04: the django-fobi==0.4.2 contains a management command which makes the necessary changes in the database for safe upgrade).

- The `captcha` field has been moved from `fobi.contrib.plugins.form_elements.fields.captcha` to `fobi.contrib.plugins.form_elements.security.captcha`. Make sure to update the package paths in `INSTALLED_APPS` of your projects' settings module (`settings.py`) when upgrading to this version.
- The `honeypot` field has been added.
- The `birthday` field has been renamed to `date_drop_down` (A real birthday field is still to come in later releases). The change causes backwards incompatibility issues if you have used that `birthday` field. If you haven't - you have nothing to worry. If you have been using it, grab the 0.3.4 version, copy the `fobi.contrib.plugins.form_elements.fields.date_drop_down` package to your project apps, make necessary path changes and update the package paths in `INSTALLED_APPS` settings module (`settings.py`) before upgrading to this version. Then, in Django admin management interface, replace all the occurrences of `Birthday` field with `Date drop down` field.
- Better error validation/handling of hidden fields. A new form snippet template added for displaying the non-field and hidden fields errors. The new template makes a part of a standard theme as an attribute `form_non_field_and_hidden_errors_snippet_template`.
- Minor fixes in generic templates.
- An additional property `is_hidden` added to the hidden form elements. Those form elements would be getting a default `TextInput` widget in the edit mode instead of the widget they come from by default. It's possible to provide an alternative widget for the edit mode as well. Default value of the `is_hidden` is set to `False`.

38.2.163 0.3.4

2014-11-23

- New settings `FOBI_FAIL_ON_ERRORS_IN_FORM_ELEMENT_PLUGINS` and `FOBI_FAIL_ON_ERRORS_IN_FORM_HANDLER_PLUGINS` introduced. They do as their name tells. Default value for both is `False`.
- Fixed exceptions raised when unicode characters were used as form names.
- Fixed exceptions raised when unicode characters were used as field labels.
- Fixes in the *db_store* and *mail* plugins related to usage of unicode characters.

38.2.164 0.3.3

2014-11-22

- Clean up the setup. Remove redundant dependencies.
- Documentation improvements.

38.2.165 0.3.2

2014-11-20

- DjangoCMS integration app made compatible with DjangoCMS 2.4.3.

38.2.166 0.3.1

2014-11-19

- DjangoCMS integration app.

38.2.167 0.3

2014-11-09

Note: Note, that this release contains minor backwards incompatible changes. The changes do not anyhow affect your existing forms or data. The only thing you need to do is update the app paths in the `settings` module of your project.

- Minor core improvements related to the theming of the form handler plugins.
- Several presentational form element plugins have been renamed. The `fobi.contrib.plugins.form_elements.content.image` plugin has been renamed to `fobi.contrib.plugins.form_elements.content.content_image`. The `fobi.contrib.plugins.form_elements.content.text` plugin has been renamed to `fobi.contrib.plugins.form_elements.content.content_text`. The `fobi.contrib.plugins.form_elements.content.video` plugin has been renamed to `fobi.contrib.plugins.form_elements.content.content_video`. If you have used any of the above mentioned plugins, make sure to update the app paths in the `settings` module of your project.

- The `fobi.contrib.plugins.form_elements.content.dummy` plugin has been moved to `fobi.contrib.plugins.form_elements.test.dummy` location. If you have used it, make sure to update the its' path in the `settings` module of your project.
- Added readme to the following content form element plugins: `dummy`, `content_image`, `content_text` and `content_video`.
- Added `foundation5` and `simple` theme widgets for `db_store` plugin.
- If you have been overriding the defaults of the `db_store` plugin, change the prefix from `FOBI_PLUGIN_DB_EXPORT_` to `FOBI_PLUGIN_DB_STORE_`. For example, `FOBI_PLUGIN_DB_EXPORT_CSV_DELIMITER` should become `FOBI_PLUGIN_DB_STORE_CSV_DELIMITER`.
- Mentioning the `fobi_find_broken_entries` management command in the documentation, as well as improving the management command itself (more verbose output).
- Birthday field added.

38.2.168 0.2.1

2014-11-06

- Minor improvements of the `db_store` plugin.
- Minor improvements of the `simple` theme. Make sure that custom form handler actions are properly shown in the form handlers list.
- Make it possible to fail silently on missing form element or form handler plugins by setting the respected values to `False`: `FOBI_FAIL_ON_MISSING_FORM_ELEMENT_PLUGINS`, `FOBI_FAIL_ON_MISSING_FORM_HANDLER_PLUGINS`. Otherwise an appropriate exception is raised.

38.2.169 0.2

2014-11-05

Note, that this release contains minor backwards incompatible changes.

- Minor (backwards incompatible) changes in the form handler plugin API. From now on both `custom_actions` and `get_custom_actions` methods accept `form_entry` (obligatory) and `request` (optional) arguments. If you have written your own or have changed existing form handler plugins with use of one of the above mentioned methods, append those arguments to the method declarations when upgrading to this version. If you haven't written your own or changed existing form handler plugins, you may just upgrade to this version.
- Added data export features to the `db_store` plugin.
- Minor fixes in `db_store` plugin.
- Added missing documentation for the `feincms_integration` app.
- Updated translations for Dutch and Russian.

38.2.170 0.1.6

2014-10-25

- Minor improvements in the theming API. From now on the `view_embed_form_entry_ajax_template` template would be used when integrating the form rendering from other products (for example, a CMS page, which has a widget which references the form object. If that property is left empty, the `view_form_entry_ajax_template` is used. For a success page the `embed_form_entry_submitted_ajax_template` template would be used.
- Functional improvements of the FeinCMS integration (the widget). If you have used the FeinCMS widget of earlier versions, you likely want to update to this one. From now on you can select a custom form title and the button text, as well as provide custom success page title and the success message; additionally, it has been made possible to hide the form- or success-page- titles.

38.2.171 0.1.5

2014-10-23

- Minor fixes in the `Integer` form element plugin.
- Minor fixes in the `Input` form element plugin.
- Minor fixes in themes (disable HTML5 form validation in edit mode).
- Minor documentation improvements.

38.2.172 0.1.4

2014-10-22

- Minor core improvements.
- Django 1.5 support improvements.
- Django 1.7 support improvements.
- Added `Captcha` form element plugin.
- Added highly-customisable `Input` form element plugin - a custom input field with support for almost any ever existing HTML attribute.
- Documentation improvements.

38.2.173 0.1.3

2014-10-13

- Django 1.7 support.
- Add HTML5 “placeholder” field attribute support.

38.2.174 0.1.2

2014-10-11

- Simple theme fixes: Fix for making the theme work in Django 1.5.

38.2.175 0.1.1

2014-10-11

- Bootstrap 3 theme fixes: When tab pane has no or little content so that the height of the dropdown menu exceeds the height of the tab pane content the dropdown menu now becomes scrollable (vertically).

38.2.176 0.1

2014-10-11

- Initial release.

38.3 Licenses

Below information about third-party packages used in the project is presented.

38.3.1 Bootstrap3

<http://getbootstrap.com/>

Licensing information:

- Apache license 2.0 (<http://www.apache.org/licenses/LICENSE-2.0>)

38.3.2 Bootstrap 3 DateTime picker

<http://eonasdan.github.io/bootstrap-datetimepicker/>

- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.3 bootstrap-slider

<https://github.com/seiyria/bootstrap-slider>

- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.4 django-autoslug

<https://bitbucket.org/neithere/django-autoslug>

Licensing information:

- LGPL 3 (<https://www.gnu.org/licenses/lgpl.html>)

38.3.5 django-formtools

<https://github.com/django/django-formtools>

Licensing information:

- BSD (<http://opensource.org/licenses/BSD-3-Clause>)

38.3.6 django-nine

<https://github.com/barseghyanartur/django-nine>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)
- LGPL 2.1 (<https://www.gnu.org/licenses/lgpl-2.1.html>)

38.3.7 django-nonefield

<https://github.com/barseghyanartur/django-nonefield>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)
- LGPL 2.1 (<https://www.gnu.org/licenses/lgpl-2.1.html>)

38.3.8 easy-thumbnails

<https://github.com/SmileyChris/easy-thumbnails>

Licensing information:

- BSD (<http://opensource.org/licenses/BSD-3-Clause>)

38.3.9 Font-awesome

<http://fontawesome.io/>

Licensing information:

- <http://fontawesome.io/license/>
- SIL OFL 1.1 (<http://scripts.sil.org/OFL>)
- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.10 Foundation 5

<http://foundation.zurb.com/>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.11 Foundation 5 DateTime picker

<https://github.com/aliibrahim/foundation-datetimepicker-rails>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.12 jQuery Colorbox

<http://www.jacklmoore.com/colorbox/>

Licensing information:

- MIT (<http://www.opensource.org/licenses/mit-license.php>)

38.3.13 jQuery

<https://jquery.org>

Licensing information:

- <https://jquery.org/license/>
- MIT (<http://github.com/jquery/jquery/blob/master/MIT-LICENSE.txt>)

38.3.14 jQueryUI

<http://jqueryui.com/>

Licensing information:

- MIT (<https://github.com/jquery/jquery-ui/blob/master/MIT-LICENSE.txt>)

38.3.15 jQuery-Slugify-Plugin

<https://github.com/pmcclhaney/jQuery-Slugify-Plugin>

Licensing information:

- BSD (<https://github.com/pmcclhaney/jQuery-Slugify-Plugin/blob/master/LICENSE>).

38.3.16 moment.js

<http://momentjs.com/>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.17 ordereddict

<https://pypi.python.org/pypi/ordereddict>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.18 Pillow

<http://python-pillow.github.io/>

Licensing information:

- Python Imaging Library license (<http://www.pythonware.com/products/pil/>)

38.3.19 Requests

<https://python-requests.org>

Licensing information:

- MIT (<http://opensource.org/licenses/mit-license.html>)

38.3.20 Six

<https://bitbucket.org/gutworth/six/>

Licensing information:

- Apache 2.0 (<http://opensource.org/licenses/Apache-2.0>)

38.3.21 Unidecode

<https://pypi.python.org/pypi/Unidecode>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)

38.3.22 vishap

<https://pypi.python.org/pypi/vishap>

Licensing information:

- GPL 2.0 (<http://www.gnu.org/licenses/gpl-2.0.html>)
- LGPL 2.1 (<https://www.gnu.org/licenses/lgpl-2.1.html>)

38.4 fobi package

38.4.1 Subpackages

38.4.1.1 fobi.contrib package

38.4.1.1.1 Subpackages

38.4.1.1.1.1 fobi.contrib.apps package

38.4.1.1.1.2 Subpackages

38.4.1.1.1.3 fobi.contrib.apps.djangocms_integration package

38.4.1.1.1.4 Submodules

38.4.1.1.1.5 fobi.contrib.apps.djangocms_integration.apps module

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

38.4.1.1.1.6 fobi.contrib.apps.djangocms_integration.cms_plugins module

38.4.1.1.1.7 fobi.contrib.apps.djangocms_integration.cms_version module

38.4.1.1.1.8 fobi.contrib.apps.djangocms_integration.conf module

```
fobi.contrib.apps.djangocms_integration.conf.get_setting(setting, override=None)
```

Get setting.

Get a setting from `fobi.contrib.apps.djangocms_integration.conf` module, falling back to the default.

If `override` is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

38.4.1.1.1.9 fobi.contrib.apps.djangocms_integration.defaults module

38.4.1.1.1.10 fobi.contrib.apps.djangocms_integration.helpers module

`fobi.contrib.apps.djangocms_integration.helpers.get_form_template_choices()`

Get the form template choices.

It's possible to provide theme templates per theme or just per project.

Return list

`fobi.contrib.apps.djangocms_integration.helpers.get_success_page_template_choices()`

Get success page template choices.

Return list

38.4.1.1.1.11 fobi.contrib.apps.djangocms_integration.models module

38.4.1.1.1.12 fobi.contrib.apps.djangocms_integration.settings module

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

38.4.1.1.1.13 Module contents

38.4.1.1.1.14 fobi.contrib.apps.drf_integration package

38.4.1.1.1.15 Subpackages

38.4.1.1.1.16 fobi.contrib.apps.drf_integration.form_elements package

38.4.1.1.1.17 Subpackages

38.4.1.1.1.18 fobi.contrib.apps.drf_integration.form_elements.content package

38.4.1.1.1.19 Subpackages

38.4.1.1.1.20 fobi.contrib.apps.drf_integration.form_elements.content.content_image package

38.4.1.1.1.21 Submodules

38.4.1.1.1.22 fobi.contrib.apps.drf_integration.form_elements.content.content_image.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image.apps.Config(app
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_image'
    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_image'
```

38.4.1.1.1.23 fobi.contrib.apps.drf_integration.form_elements.content.content_image.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image.base.ContentImageBase
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None, form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_image'
```

38.4.1.1.1.24 fobi.contrib.apps.drf_integration.form_elements.content.content_image.fobi_integration_form_element.module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image.fobi_integration_form_element.FobiIntegrationFormElement
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None, form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_image'
```

38.4.1.1.1.25 Module contents

38.4.1.1.1.26 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url package

38.4.1.1.1.27 Submodules

38.4.1.1.1.28 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.apps.Config
    Bases: django.apps.config.AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_image_url'
    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_image_url'
```

38.4.1.1.1.29 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.base.ContentImageURLFormElementPlugin
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None, form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_image_url'
```

38.4.1.1.1.30 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.fobi_integration_form_element.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_image_url.fobi_integration_form_element.base.FobiIntegrationFormElementPlugin
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None, form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_image_url'
```

38.4.1.1.1.31 Module contents

38.4.1.1.1.32 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown package

38.4.1.1.1.33 Submodules

38.4.1.1.1.34 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.apps.Config
    Bases: django.apps.config.AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_markdown'
```

```
name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_markdown'
```

38.4.1.1.1.35 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.base.Content
    Bases:      fobi.base.IntegrationFormElementPlugin,      fobi.contrib.apps.
               drf_integration.base.DRFSubmitPluginFormDataMixin

    Content markdown (CharField) plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_markdown'
```

38.4.1.1.1.36 fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.fobi_integration_form_e module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_markdown.fobi_integrat
    Bases:      fobi.base.IntegrationFormElementPlugin,      fobi.contrib.apps.
               drf_integration.base.DRFSubmitPluginFormDataMixin

    Content markdown (CharField) plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_markdown'
```

38.4.1.1.1.37 Module contents

38.4.1.1.1.38 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext package

38.4.1.1.1.39 Submodules

38.4.1.1.1.40 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.apps.Config

    Bases: django.apps.config.AppConfig
```

Config.

```
label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_richtext'
name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_richtext'
```

38.4.1.1.1.41 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base.ContentRichTextPlugin
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    Content rich text (CharField) plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,
                                form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_richtext'
```

38.4.1.1.1.42 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.fobi_integration_form_elements.ContentRichTextPlugin
    Bases: fobi.base.IntegrationFormElementPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    Content rich text (CharField) plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,
                                form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_richtext'
```

38.4.1.1.1.43 Module contents

38.4.1.1.1.44 fobi.contrib.apps.drf_integration.form_elements.content.content_text package

38.4.1.1.1.45 Submodules

38.4.1.1.1.46 fobi.contrib.apps.drf_integration.form_elements.content.content_text.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_text.apps.Config(app_
    app_
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_text'
    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_text'
```

38.4.1.1.1.47 fobi.contrib.apps.drf_integration.form_elements.content.content_text.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_text.base.ContentText
    Bases:      fobi.base.IntegrationFormElementPlugin,      fobi.contrib.apps.
               drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_text'
```

38.4.1.1.1.48 fobi.contrib.apps.drf_integration.form_elements.content.content_text.fobi_integration_form_element module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_text.fobi_integration
    Bases:      fobi.base.IntegrationFormElementPlugin,      fobi.contrib.apps.
               drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, has_value=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'content_text'
```


38.4.1.1.1.49 Module contents

38.4.1.1.1.50 fobi.contrib.apps.drf_integration.form_elements.content.content_video package

38.4.1.1.1.51 Submodules

38.4.1.1.1.52 fobi.contrib.apps.drf_integration.form_elements.content.content_video.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_video.apps.Config(app
    app

    Bases: django.apps.config AppConfig

    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_content_content_video'

    name = 'fobi.contrib.apps.drf_integration.form_elements.content.content_video'
```

38.4.1.1.1.53 fobi.contrib.apps.drf_integration.form_elements.content.content_video.base module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_video.base.ContentVi
    Bases:      fobi.base.IntegrationFormElementPlugin,      fobi.contrib.apps.
    drf_integration.base.DRFSubmitPluginFormDataMixin

    CharField plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, has_value=None, **kwargs)

        Get form field instances.

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

    integrate_with = 'drf_integration'

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

    uid = 'content_video'
```

38.4.1.1.1.54 fobi.contrib.apps.drf_integration.form_elements.content.content_video.fobi_integration_form_element module

```
class fobi.contrib.apps.drf_integration.form_elements.content.content_video.fobi_integrati
    Bases:      fobi.base.IntegrationFormElementPlugin,      fobi.contrib.apps.
    drf_integration.base.DRFSubmitPluginFormDataMixin

    CharField plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, has_value=None, **kwargs)

        Get form field instances.

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

    integrate_with = 'drf_integration'

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

    uid = 'content_video'
```

38.4.1.1.1.55 Module contents

38.4.1.1.1.56 Module contents

38.4.1.1.1.57 fobi.contrib.apps.drf_integration.form_elements.fields package

38.4.1.1.1.58 Subpackages

38.4.1.1.1.59 fobi.contrib.apps.drf_integration.form_elements.fields.boolean package

38.4.1.1.1.60 Submodules

38.4.1.1.1.61 fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_boolean'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.boolean'
```

38.4.1.1.1.62 fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.boolean.base.BooleanSelectPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    Boolean select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'boolean'
```

38.4.1.1.1.63 fobi.contrib.apps.drf_integration.form_elements.fields.boolean.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.boolean.fobi_integration_form_
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    Boolean select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
```

```
group = <django.utils.functional.lazy.<locals>.__proxy__ object>
integrate_with = 'drf_integration'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'boolean'
```

38.4.1.1.164 Module contents

38.4.1.1.165 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple package

38.4.1.1.166 Submodules

38.4.1.1.167 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.apps
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_checkbox_select_multip
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multipl
```

38.4.1.1.168 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.base
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    MultipleChoiceField plugin.
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    submit_plugin_form_data (form_element_plugin, form_entry, request, serializer,
                            form_element_entries=None, **kwargs)
        Submit plugin form data.
        Called on form submission (when user actually posts the data to assembled form).
```

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.

- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'checkbox_select_multiple'

38.4.1.1.1.69 fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.fobi_integration_form module

class fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.fobi_integration_form

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

MultipleChoiceField plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'checkbox_select_multiple'

38.4.1.1.1.70 Module contents

38.4.1.1.1.71 fobi.contrib.apps.drf_integration.form_elements.fields.date package

38.4.1.1.1.72 Submodules

38.4.1.1.1.73 fobi.contrib.apps.drf_integration.form_elements.fields.date.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.date.apps.**Config** (*app_name*, *app_module*)

Bases: *django.apps.config.AppConfig*

Config.

```
label = 'fobi_contrib_apps_drf_integration_form_elements_fields_date'
name = 'fobi.contrib.apps.drf_integration.form_elements.fields.date'
```

38.4.1.1.1.74 fobi.contrib.apps.drf_integration.form_elements.fields.date.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date.base.DateSelectPlugin (use
    Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin

    DateSelect select plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'date'
```

38.4.1.1.1.75 fobi.contrib.apps.drf_integration.form_elements.fields.date.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date.fobi_integration_form_el
    Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin

    DateSelect select plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'date'
```

38.4.1.1.1.76 Module contents

38.4.1.1.1.77 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down package

38.4.1.1.1.78 Submodules

38.4.1.1.1.79 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.apps.Config (app
    Bases: django.apps.config.AppConfig
    Config.
```

```
label = 'fobi_contrib_apps_drf_integration_form_elements_fields_date_drop_down'
name = 'fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down'
```

38.4.1.1.1.80 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.base.DateDropDown
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    DateSelect select plugin.
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'date_drop_down'
```

38.4.1.1.1.81 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down.fobi_integration_form_elements
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    DateSelect select plugin.
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'date_drop_down'
```

38.4.1.1.1.82 Module contents

38.4.1.1.1.83 fobi.contrib.apps.drf_integration.form_elements.fields.datetime package

38.4.1.1.1.84 Submodules

38.4.1.1.1.85 fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.datetime.apps.Config (app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
```

```
label = 'fobi_contrib_apps_drf_integration_form_elements_fields_datetime'
name = 'fobi.contrib.apps.drf_integration.form_elements.fields.datetime'
```

38.4.1.1.1.86 fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base.DateTimeSelectPlugin
    Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
                drf_integration.base.DRFSubmitPluginFormDataMixin

    DateTimeSelect select plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'datetime'
```

38.4.1.1.1.87 fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_integration_form_elements
    Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
                drf_integration.base.DRFSubmitPluginFormDataMixin

    DateTimeSelect select plugin.

    get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'datetime'
```

38.4.1.1.1.88 Module contents

38.4.1.1.1.89 fobi.contrib.apps.drf_integration.form_elements.fields.decimal package

38.4.1.1.1.90 Submodules

38.4.1.1.1.91 fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.decimal.apps.Config (app_name,
                                                                                      app_module)
    Bases: django.apps.config.AppConfig
    Config.
```

```
label = 'fobi_contrib_apps_drf_integration_form_elements_fields_decimal'
name = 'fobi.contrib.apps.drf_integration.form_elements.fields.decimal'
```

38.4.1.1.1.92 fobi.contrib.apps.drf_integration.form_elements.fields.decimal.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.decimal.base.DecimalInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    DecimalField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'decimal'
```

38.4.1.1.1.93 fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi_integration_form_
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    DecimalField plugin.

    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'decimal'
```

38.4.1.1.1.94 Module contents

38.4.1.1.1.95 fobi.contrib.apps.drf_integration.form_elements.fields.duration package

38.4.1.1.1.96 Submodules

38.4.1.1.1.97 fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.duration.apps.Config(app_name,
                                                                                     app_module)
    Bases: django.apps.config.AppConfig
    Config.
```



```
label = 'fobi_contrib_apps_drf_integration_form_elements_fields_duration'
name = 'fobi.contrib.apps.drf_integration.form_elements.fields.duration'
```

38.4.1.1.1.98 fobi.contrib.apps.drf_integration.form_elements.fields.duration.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.duration.base.DurationInputPl
Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin
```

DurationField plugin.

```
get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                             form_element_entries=None, **kwargs)
```

Get form field instances.

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

```
integrate_with = 'drf_integration'
```

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

```
submit_plugin_form_data (form_element_plugin,      form_entry,      request,      serializer,
                           form_element_entries=None, **kwargs)
```

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

```
uid = 'duration'
```

38.4.1.1.1.99 fobi.contrib.apps.drf_integration.form_elements.fields.duration.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.duration.fobi_integration_for
Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin
```

DurationField plugin.

```
get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                             form_element_entries=None, **kwargs)
```

Get form field instances.

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

```
integrate_with = 'drf_integration'
```

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

submit_plugin_form_data (*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'duration'

38.4.1.1.1.100 Module contents

38.4.1.1.1.101 *fobi.contrib.apps.drf_integration.form_elements.fields.email* package

38.4.1.1.1.102 Submodules

38.4.1.1.1.103 *fobi.contrib.apps.drf_integration.form_elements.fields.email.apps* module

class *fobi.contrib.apps.drf_integration.form_elements.fields.email.apps.Config* (*app_name, app_module*)

Bases: *django.apps.config AppConfig*

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_email'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.email'

38.4.1.1.1.104 *fobi.contrib.apps.drf_integration.form_elements.fields.email.base* module

class *fobi.contrib.apps.drf_integration.form_elements.fields.email.base.EmailInputPlugin* (*us*)

Bases: *fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

EmailField plugin.

get_custom_field_instances (*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

uid = 'email'

38.4.1.1.1.105 fobi.contrib.apps.drf_integration.form_elements.fields.email.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.email.fobi_integration_form_e
Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin

EmailField plugin.

get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                             form_element_entries=None, **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
integrate_with = 'drf_integration'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'email'
```

38.4.1.1.1.106 Module contents

38.4.1.1.1.107 fobi.contrib.apps.drf_integration.form_elements.fields.file package

38.4.1.1.1.108 Submodules

38.4.1.1.1.109 fobi.contrib.apps.drf_integration.form_elements.fields.file.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.file.apps.Config (app_name,
                                     app_module)
Bases: django.apps.config.AppConfig
Config.
label = 'fobi_contrib_apps_drf_integration_form_elements_fields_file'
name = 'fobi.contrib.apps.drf_integration.form_elements.fields.file'
```

38.4.1.1.1.110 fobi.contrib.apps.drf_integration.form_elements.fields.file.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.file.base.FileInputPlugin (user=
Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin

FormField plugin.

get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                             form_element_entries=None, **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
integrate_with = 'drf_integration'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

submit_plugin_form_data (*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'file'

38.4.1.1.111 *fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_integration_form_elements* module

class *fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_integration_form_el*
 Bases: *fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

FileField plugin.

get_custom_field_instances (*form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'file'

38.4.1.1.1.112 Module contents

38.4.1.1.1.113 fobi.contrib.apps.drf_integration.form_elements.fields.float package

38.4.1.1.1.114 Submodules

38.4.1.1.1.115 fobi.contrib.apps.drf_integration.form_elements.fields.float.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.float.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_float'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.float'
```

38.4.1.1.1.116 fobi.contrib.apps.drf_integration.form_elements.fields.float.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.float.base.FloatInputPlugin(us
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin
    FloatField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'float'
```

38.4.1.1.1.117 fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin
    FloatField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'float'
```

38.4.1.1.1.118 Module contents

38.4.1.1.1.119 fobi.contrib.apps.drf_integration.form_elements.fields.hidden package

38.4.1.1.1.120 Submodules

38.4.1.1.1.121 fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_hidden'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.hidden'
```

38.4.1.1.1.122 fobi.contrib.apps.drf_integration.form_elements.fields.hidden.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.hidden.base.HiddenInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'hidden'
```

38.4.1.1.1.123 fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'hidden'
```

38.4.1.1.1.124 Module contents

38.4.1.1.1.125 fobi.contrib.apps.drf_integration.form_elements.fields.input package

38.4.1.1.1.126 Submodules

38.4.1.1.1.127 fobi.contrib.apps.drf_integration.form_elements.fields.input.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.input.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_input'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.input'
```

38.4.1.1.1.128 fobi.contrib.apps.drf_integration.form_elements.fields.input.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.input.base.InputPlugin(user=None)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'input'
```

38.4.1.1.1.129 fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    CharField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'input'
```

38.4.1.1.1.130 Module contents

38.4.1.1.1.131 fobi.contrib.apps.drf_integration.form_elements.fields.integer package

38.4.1.1.1.132 Submodules

38.4.1.1.1.133 fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_integer'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.integer'
```

38.4.1.1.1.134 fobi.contrib.apps.drf_integration.form_elements.fields.integer.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.integer.base.IntegerInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    IntegerField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'integer'
```

38.4.1.1.1.135 fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_integration_form_
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
            drf_integration.base.DRFSubmitPluginFormDataMixin
    IntegerField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'integer'
```


38.4.1.1.1.136 Module contents

38.4.1.1.1.137 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address package

38.4.1.1.1.138 Submodules

38.4.1.1.1.139 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps.Config(app_name, app_model)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_ip_address'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.ip_address'
```

38.4.1.1.1.140 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base.IPAddressInputFormPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    IPAddressField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'ip_address'
```

38.4.1.1.1.141 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration_form_elements.FormPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    IPAddressField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'ip_address'
```

38.4.1.1.1.142 Module contents

38.4.1.1.1.143 fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean package

38.4.1.1.1.144 Submodules

38.4.1.1.1.145 fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps.Config(app_name, app_name)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_null_boolean'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean'
```

38.4.1.1.1.146 fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.base.NullBooleanField
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    NullBooleanField select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'null_boolean'
```

38.4.1.1.1.147 fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.fobi_integration_form_elements.NullBooleanField
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    NullBooleanField select plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'null_boolean'
```

38.4.1.1.1.148 Module contents

38.4.1.1.1.149 fobi.contrib.apps.drf_integration.form_elements.fields.password package

38.4.1.1.1.150 Submodules

38.4.1.1.1.151 fobi.contrib.apps.drf_integration.form_elements.fields.password.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.password.apps.Config(app_name,  
                                     app_module)  
    Bases: django.apps.config.AppConfig  
    Config.  
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_password'  
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.password'
```

38.4.1.1.1.152 fobi.contrib.apps.drf_integration.form_elements.fields.password.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.password.base.PasswordInputPl  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.  
           drf_integration.base.DRFSubmitPluginFormDataMixin  
    CharField plugin.  
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,  
                               form_element_entries=None, **kwargs)  
        Get form field instances.  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'password'
```

38.4.1.1.1.153 fobi.contrib.apps.drf_integration.form_elements.fields.password.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.password.fobi_integration_for  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.  
           drf_integration.base.DRFSubmitPluginFormDataMixin  
    CharField plugin.  
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,  
                               form_element_entries=None, **kwargs)  
        Get form field instances.  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'password'
```

38.4.1.1.1.154 Module contents

38.4.1.1.1.155 fobi.contrib.apps.drf_integration.form_elements.fields.radio package

38.4.1.1.1.156 Submodules

38.4.1.1.1.157 fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_radio'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.radio'
```

38.4.1.1.1.158 fobi.contrib.apps.drf_integration.form_elements.fields.radio.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.radio.base.RadioInputPlugin(us
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'radio'
```

38.4.1.1.1.159 fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                               form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'radio'
```

38.4.1.1.1.160 Module contents

38.4.1.1.1.161 fobi.contrib.apps.drf_integration.form_elements.fields.range_select package

38.4.1.1.1.162 Submodules

38.4.1.1.1.163 fobi.contrib.apps.drf_integration.form_elements.fields.range_select.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.range_select.apps.Config(app_name, app_label)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_range_select'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.range_select'
```

38.4.1.1.1.164 fobi.contrib.apps.drf_integration.form_elements.fields.range_select.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.range_select.base.RangeSelectBase(form_element_plugin)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'range_select'
```

38.4.1.1.1.165 fobi.contrib.apps.drf_integration.form_elements.fields.range_select.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.range_select.fobi_integration_form_elements.RangeSelect(form_element_plugin)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'range_select'
```

38.4.1.1.1.166 Module contents

38.4.1.1.1.167 fobi.contrib.apps.drf_integration.form_elements.fields.regex package

38.4.1.1.1.168 Submodules

38.4.1.1.1.169 fobi.contrib.apps.drf_integration.form_elements.fields.regex.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.regex.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_regex'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.regex'
```

38.4.1.1.1.170 fobi.contrib.apps.drf_integration.form_elements.fields.regex.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.regex.base.RegexInputPlugin(us)
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    RegexField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'regex'
```

38.4.1.1.1.171 fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    RegexField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'regex'
```

38.4.1.1.1.172 Module contents

38.4.1.1.1.173 fobi.contrib.apps.drf_integration.form_elements.fields.select package

38.4.1.1.1.174 Submodules

38.4.1.1.1.175 fobi.contrib.apps.drf_integration.form_elements.fields.select.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select'
```

38.4.1.1.1.176 fobi.contrib.apps.drf_integration.form_elements.fields.select.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select.base.SelectInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'select'
```

38.4.1.1.1.177 fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.
           drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None,
                              form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'select'
```

38.4.1.1.1.178 Module contents

38.4.1.1.1.179 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object package

38.4.1.1.1.180 Submodules

38.4.1.1.1.181 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.apps.**Conf**

Bases: django.apps.config.AppConfig

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select_model_object'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object'

38.4.1.1.1.182 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.base.**Select**

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

ChoiceField plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_model_object'

38.4.1.1.1.183 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.fobi_integration_form_e module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object.fobi_integ

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

ChoiceField plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_model_object'

38.4.1.1.1.184 Module contents

38.4.1.1.1.185 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple package

38.4.1.1.1.186 Submodules

38.4.1.1.1.187 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.apps.**Config** (*ap*

Bases: *django.apps.config.AppConfig*

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select_multiple'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple'

38.4.1.1.1.188 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base.**SelectMu**

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

MultipleChoiceField plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_multiple'

38.4.1.1.1.189 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.fobi_integration_form_element module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.fobi_integrat

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

MultipleChoiceField plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_multiple'

38.4.1.1.1.190 Module contents

38.4.1.1.1.191 *fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects* package

38.4.1.1.1.192 Submodules

38.4.1.1.1.193 *fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.apps* module

class *fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects*

Bases: *django.apps.config.AppConfig*

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select_multiple_model_'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_ol'

38.4.1.1.1.194 *fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.base* module

class *fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects*

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

ModelMultipleChoiceField plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_multiple_model_objects'

38.4.1.1.1.195 *fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects.fobi_integration.module*

class *fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_model_objects*

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

ModelMultipleChoiceField plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_multiple_model_objects'

38.4.1.1.1.196 Module contents

38.4.1.1.1.197 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max package

38.4.1.1.1.198 Submodules

38.4.1.1.1.199 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.apps
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_select_multiple_with_max'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max'
```

38.4.1.1.1.200 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.base
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    MultipleChoiceField with max plugin.
    get_custom_field_instances(form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    submit_plugin_form_data(form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs)
        Submit plugin form data.
        Called on form submission (when user actually posts the data to assembled form).
    Parameters
        • form_element_plugin –
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • serializer (rest_framework.serializers.Serializer) –
        • form_element_entries (iterable) –
    uid = 'select_multiple_with_max'
```

38.4.1.1.1.201 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.fobi_integration_f module

class fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max.fobi

Bases: *fobi.base.IntegrationFormFieldPlugin*, *fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin*

MultipleChoiceField with max plugin.

get_custom_field_instances (*form_element_plugin*, *request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.

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

integrate_with = 'drf_integration'

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

submit_plugin_form_data (*form_element_plugin*, *form_entry*, *request*, *serializer*, *form_element_entries=None*, ***kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_element_plugin** –
- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

uid = 'select_multiple_with_max'

38.4.1.1.1.202 Module contents

38.4.1.1.1.203 fobi.contrib.apps.drf_integration.form_elements.fields.slider package

38.4.1.1.1.204 Submodules

38.4.1.1.1.205 fobi.contrib.apps.drf_integration.form_elements.fields.slider.apps module

class fobi.contrib.apps.drf_integration.form_elements.fields.slider.apps.**Config** (*app_name*, *app_module*)

Bases: *django.apps.config.AppConfig*

Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_slider'

name = 'fobi.contrib.apps.drf_integration.form_elements.fields.slider'

38.4.1.1.1.206 fobi.contrib.apps.drf_integration.form_elements.fields.slider.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slider.base.SliderInputPlugin
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'slider'
```

38.4.1.1.1.207 fobi.contrib.apps.drf_integration.form_elements.fields.slider.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slider.fobi_integration_form_e
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    ChoiceField plugin.

    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'slider'
```

38.4.1.1.1.208 Module contents

38.4.1.1.1.209 fobi.contrib.apps.drf_integration.form_elements.fields.slug package

38.4.1.1.1.210 Submodules

38.4.1.1.1.211 fobi.contrib.apps.drf_integration.form_elements.fields.slug.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slug.apps.Config (app_name,
                                                                              app_module)
    Bases: django.apps.config.AppConfig
    Config.

    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_slug'
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.slug'
```

38.4.1.1.1.212 fobi.contrib.apps.drf_integration.form_elements.fields.slug.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slug.base.SlugInputPlugin (user=  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.  
drf_integration.base.DRFSubmitPluginFormDataMixin  
  
    SlugField plugin.  
  
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,  
                                form_element_entries=None, **kwargs)  
        Get form field instances.  
  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'slug'
```

38.4.1.1.1.213 fobi.contrib.apps.drf_integration.form_elements.fields.slug.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.slug.fobi_integration_form_el  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.  
drf_integration.base.DRFSubmitPluginFormDataMixin  
  
    SlugField plugin.  
  
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,  
                                form_element_entries=None, **kwargs)  
        Get form field instances.  
  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'slug'
```

38.4.1.1.1.214 Module contents

38.4.1.1.1.215 fobi.contrib.apps.drf_integration.form_elements.fields.text package

38.4.1.1.1.216 Submodules

38.4.1.1.1.217 fobi.contrib.apps.drf_integration.form_elements.fields.text.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.text.apps.Config (app_name,  
                                     app_module)  
  
    Bases: django.apps.config.AppConfig  
    Config.  
  
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_text'  
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.text'
```


38.4.1.1.1.218 fobi.contrib.apps.drf_integration.form_elements.fields.text.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.text.base.TextInputPlugin (user=  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin  
    CharField plugin.  
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,  
                                form_element_entries=None, **kwargs)  
        Get form field instances.  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'text'
```

38.4.1.1.1.219 fobi.contrib.apps.drf_integration.form_elements.fields.text.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.text.fobi_integration_form_el  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin  
    CharField plugin.  
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None,  
                                form_element_entries=None, **kwargs)  
        Get form field instances.  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'text'
```

38.4.1.1.1.220 Module contents

38.4.1.1.1.221 fobi.contrib.apps.drf_integration.form_elements.fields.textarea package

38.4.1.1.1.222 Submodules

38.4.1.1.1.223 fobi.contrib.apps.drf_integration.form_elements.fields.textarea.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.textarea.apps.Config (app_name,  
                                           app_module  
    Bases: django.apps.config.AppConfig  
    Config.  
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_textarea'  
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.textarea'
```

38.4.1.1.1.224 fobi.contrib.apps.drf_integration.form_elements.fields.textarea.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.textarea.base.TextareaPlugin (  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin  
  
    CharField plugin.  
  
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)  
        Get form field instances.  
  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'textarea'
```

38.4.1.1.1.225 fobi.contrib.apps.drf_integration.form_elements.fields.textarea.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.textarea.fobi_integration_form_elements.FobiIntegrationFormElementsPlugin (  
    Bases: fobi.base.IntegrationFormFieldPlugin, fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin  
  
    CharField plugin.  
  
    get_custom_field_instances (form_element_plugin, request=None, form_entry=None, form_element_entries=None, **kwargs)  
        Get form field instances.  
  
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    integrate_with = 'drf_integration'  
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>  
    uid = 'textarea'
```

38.4.1.1.1.226 Module contents

38.4.1.1.1.227 fobi.contrib.apps.drf_integration.form_elements.fields.time package

38.4.1.1.1.228 Submodules

38.4.1.1.1.229 fobi.contrib.apps.drf_integration.form_elements.fields.time.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.time.apps.Config (app_name, app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    Config.  
  
    label = 'fobi_contrib_apps_drf_integration_form_elements_fields_time'  
    name = 'fobi.contrib.apps.drf_integration.form_elements.fields.time'
```

38.4.1.1.1.230 fobi.contrib.apps.drf_integration.form_elements.fields.time.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.time.base.TimeInputPlugin (user=
Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin

TimeField plugin.

get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                             form_element_entries=None, **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
integrate_with = 'drf_integration'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'time'
```

38.4.1.1.1.231 fobi.contrib.apps.drf_integration.form_elements.fields.time.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.time.fobi_integration_form_el
Bases:      fobi.base.IntegrationFormFieldPlugin,      fobi.contrib.apps.
drf_integration.base.DRFSubmitPluginFormDataMixin

TimeField plugin.

get_custom_field_instances (form_element_plugin,      request=None,      form_entry=None,
                             form_element_entries=None, **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
integrate_with = 'drf_integration'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'time'
```

38.4.1.1.1.232 Module contents

38.4.1.1.1.233 fobi.contrib.apps.drf_integration.form_elements.fields.url package

38.4.1.1.1.234 Submodules

38.4.1.1.1.235 fobi.contrib.apps.drf_integration.form_elements.fields.url.apps module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.url.apps.Config (app_name,
                                     app_module)
Bases: django.apps.config.AppConfig
Config.

label = 'fobi_contrib_apps_drf_integration_form_elements_fields_url'
name = 'fobi.contrib.apps.drf_integration.form_elements.fields.url'
```

38.4.1.1.1.236 fobi.contrib.apps.drf_integration.form_elements.fields.url.base module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.url.base.URLInputPlugin (user=None)
    Bases:          fobi.base.IntegrationFormFieldPlugin,          fobi.contrib.apps.
                    drf_integration.base.DRFSubmitPluginFormDataMixin
    URLField plugin.

    get_custom_field_instances (form_element_plugin,    request=None,    form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'url'
```

38.4.1.1.1.237 fobi.contrib.apps.drf_integration.form_elements.fields.url.fobi_integration_form_elements module

```
class fobi.contrib.apps.drf_integration.form_elements.fields.url.fobi_integration_form_ele
    Bases:          fobi.base.IntegrationFormFieldPlugin,          fobi.contrib.apps.
                    drf_integration.base.DRFSubmitPluginFormDataMixin
    URLField plugin.

    get_custom_field_instances (form_element_plugin,    request=None,    form_entry=None,
                                form_element_entries=None, **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'url'
```

38.4.1.1.1.238 Module contents

38.4.1.1.1.239 Module contents

38.4.1.1.1.240 Module contents

38.4.1.1.1.241 fobi.contrib.apps.drf_integration.form_handlers package

38.4.1.1.1.242 Subpackages

38.4.1.1.1.243 fobi.contrib.apps.drf_integration.form_handlers.db_store package

38.4.1.1.1.244 Submodules

38.4.1.1.1.245 fobi.contrib.apps.drf_integration.form_handlers.db_store.apps module

```
class fobi.contrib.apps.drf_integration.form_handlers.db_store.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_handlers_db_store'
    name = 'fobi.contrib.apps.drf_integration.form_handlers.db_store'
```

38.4.1.1.1.246 fobi.contrib.apps.drf_integration.form_handlers.db_store.base module

```
class fobi.contrib.apps.drf_integration.form_handlers.db_store.base.DBStoreHandlerPlugin(us)
    Bases: fobi.base.IntegrationFormHandlerPlugin
    DB store form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'db_store'
```

38.4.1.1.1.247 fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration_form_handlers module

```
class fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration_form_handl
    Bases: fobi.base.IntegrationFormHandlerPlugin
    DB store form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
```

```

name = <django.utils.functional.lazy.<locals>.__proxy__ object>
run (form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
    Run.
uid = 'db_store'

```

38.4.1.1.1.248 Module contents

38.4.1.1.1.249 fobi.contrib.apps.drf_integration.form_handlers.http_repost package

38.4.1.1.1.250 Submodules

38.4.1.1.1.251 fobi.contrib.apps.drf_integration.form_handlers.http_repost.apps module

```

class fobi.contrib.apps.drf_integration.form_handlers.http_repost.apps.Config(app_name,
                                                                              app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_handlers_http_repost'
    name = 'fobi.contrib.apps.drf_integration.form_handlers.http_repost'

```

38.4.1.1.1.252 fobi.contrib.apps.drf_integration.form_handlers.http_repost.base module

```

class fobi.contrib.apps.drf_integration.form_handlers.http_repost.base.HTTPRepostHandlerPlugin
    Bases: fobi.base.IntegrationFormHandlerPlugin
    HTTP re-post handler plugin.
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    run (form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'http_repost'

```

38.4.1.1.1.253 fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handlers module

```

class fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handlers
    Bases: fobi.base.IntegrationFormHandlerPlugin
    HTTP re-post handler plugin.
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    run (form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'http_repost'

```

38.4.1.1.1.254 Module contents

38.4.1.1.1.255 fobi.contrib.apps.drf_integration.form_handlers.mail package

38.4.1.1.1.256 Submodules

38.4.1.1.1.257 fobi.contrib.apps.drf_integration.form_handlers.mail.apps module

```
class fobi.contrib.apps.drf_integration.form_handlers.mail.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_apps_drf_integration_form_handlers_mail'
    name = 'fobi.contrib.apps.drf_integration.form_handlers.mail'
```

38.4.1.1.1.258 fobi.contrib.apps.drf_integration.form_handlers.mail.base module

```
class fobi.contrib.apps.drf_integration.form_handlers.mail.base.MailHandlerPlugin(user=None)
    Bases: fobi.base.IntegrationFormHandlerPlugin
    Mail handler form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'mail'
```

38.4.1.1.1.259 fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_integration_form_handlers module

```
class fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_integration_form_handlers.FobiIntegrationFormHandlerPlugin
    Bases: fobi.base.IntegrationFormHandlerPlugin
    Mail handler form handler plugin.
    Can be used only once per form.
    integrate_with = 'drf_integration'
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    run(form_handler_plugin, form_entry, request, form_element_entries=None, **kwargs)
        Run.
    uid = 'mail'
```

38.4.1.1.1.260 Module contents

38.4.1.1.1.261 Module contents

38.4.1.1.1.262 Submodules

38.4.1.1.1.263 fobi.contrib.apps.drf_integration.apps module

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

38.4.1.1.1.264 fobi.contrib.apps.drf_integration.base module

```
class fobi.contrib.apps.drf_integration.base.DRFIntegrationFormElementPluginProcessor(*args, **kwargs)
    Bases: fobi.base.IntegrationFormElementPluginProcessor
    Django REST framework field instance processor.
    process_custom_form_field_instance(form_element_entry, form_entry, request, form_element_plugin=None)
        Process.
class fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormDataMixin
    Bases: object
    Submit plugin form data mixin.
    submit_plugin_form_data(form_element_plugin, form_entry, request, serializer, form_element_entries=None, **kwargs)
        Submit plugin form data.
        Called on form submission (when user actually posts the data to assembled form).
```

Parameters

- **form_element_plugin** –
- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- **form_element_entries** (`iterable`) –

```
fobi.contrib.apps.drf_integration.base.get_cleaned_data(serializer, keys_to_remove=[], values_to_remove=[])
    Get cleaned data.
```

Gets cleaned data, having the trash (fields without values) filtered out.

Parameters

- **serializer** –
- **keys_to_remove** (*iterable*) –
- **values_to_remove** (*iterable*) –

Return dict

`fobi.contrib.apps.drf_integration.base.get_field_name_to_label_map(serializer, keys_to_remove=[], values_to_remove=[])`

Get field name to label map.

Parameters

- **serializer** –
- **keys_to_remove** (*iterable*) –
- **values_to_remove** (*iterable*) –

Return dict

`fobi.contrib.apps.drf_integration.base.get_processed_serializer_data(serializer, form_element_entries)`

Gets processed serializer data.

Simply fires both `fobi.base.get_cleaned_data` and `fobi.base.get_field_name_to_label_map` functions and returns the result.

Parameters

- **serializer** –
- **form_element_entries** (*iterable*) – Iterable of form element entries.

Return tuple

`fobi.contrib.apps.drf_integration.base.run_form_handlers(form_entry, request, serializer, form_element_entries=None)`

Run form handlers.

Parameters

- **form_entry** (`fobi.models.FormEntry`) –
- **request** (`django.http.HttpRequest`) –
- **serializer** (`rest_framework.serializers.Serializer`) –
- **form_element_entries** (*iterable*) –

Return tuple List of success responses, list of error responses

`fobi.contrib.apps.drf_integration.base.submit_plugin_form_data(form_entry, request, serializer, form_element_entries=None, **kwargs)`

Submit plugin form data for all plugins.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –

- **serializer** (*rest_framework.serializers.Serializer*) –
- **form_element_entries** (*iterable*) –

38.4.1.1.1.265 fobi.contrib.apps.drf_integration.dynamic module

`fobi.contrib.apps.drf_integration.dynamic.assemble_serializer_class` (*form_entry*, *request=None*, *origin=None*, *origin_kwargs_update_func=None*, *origin_return_func=None*, *form_element_entries=None*, *has_value=None*, *declared_fields=None*, *declared_fields_metadata=None*)

Assemble a serializer class by given entry.

Parameters

- **form_entry** –
- **base_class** –
- **request** (*django.http.HttpRequest*) –
- **origin** (*string*) –
- **origin_kwargs_update_func** (*callable*) –
- **origin_return_func** (*callable*) –
- **form_element_entries** (*iterable*) – If given, used instead of `form_entry`. `formelemententry_set.all` (no additional database hit).
- **has_value** (*bool*) –

`fobi.contrib.apps.drf_integration.dynamic.get_declared_fields` (*form_entry*, *request=None*, *origin=None*, *origin_kwargs_update_func=None*, *origin_return_func=None*, *form_element_entries=None*, *has_value=None*)

Get declared fields.

38.4.1.1.1.266 fobi.contrib.apps.drf_integration.fields module

class `fobi.contrib.apps.drf_integration.fields.ContentImageField` (***kwargs*)

Bases: `fobi.contrib.apps.drf_integration.fields.NoneField`

Content image field.

```
class fobi.contrib.apps.drf_integration.fields.ContentMarkdownField(**kwargs)
    Bases: fobi.contrib.apps.drf_integration.fields.NoneField

    Content markdown field.
```

```
class fobi.contrib.apps.drf_integration.fields.ContentRichTextField(**kwargs)
    Bases: fobi.contrib.apps.drf_integration.fields.NoneField

    Content rich text field.
```

```
class fobi.contrib.apps.drf_integration.fields.ContentTextField(**kwargs)
    Bases: fobi.contrib.apps.drf_integration.fields.NoneField

    Content text field.
```

```
class fobi.contrib.apps.drf_integration.fields.ContentVideoField(**kwargs)
    Bases: fobi.contrib.apps.drf_integration.fields.NoneField

    Content video field.
```

```
class fobi.contrib.apps.drf_integration.fields.ModelChoiceField(*args,
                                                                **kwargs)
    Bases: rest_framework.fields.ChoiceField, fobi.contrib.apps.drf_integration.
           fields.ModelChoiceFieldMixin

    Model choice field.
```

```
class fobi.contrib.apps.drf_integration.fields.ModelMultipleChoiceField(*args,
                                                                           **kwargs)
    Bases: rest_framework.fields.MultipleChoiceField, fobi.contrib.apps.
           drf_integration.fields.ModelChoiceFieldMixin

    Model choice field.
```

```
class fobi.contrib.apps.drf_integration.fields.MultipleChoiceWithMaxField(*args,
                                                                              **kwargs)
    Bases: rest_framework.fields.MultipleChoiceField

    MultipleChoiceWithMaxField.
```

```
default_error_messages = {'empty': <django.utils.functional.lazy.<locals>.__proxy__ of
                                <django.utils.functional.lazy object at 0x...>.>

to_internal_value(data)
    Transform the incoming primitive data into a native value.
```

```
class fobi.contrib.apps.drf_integration.fields.NoneField(**kwargs)
    Bases: rest_framework.fields.Field

    NoneField.
```

```
default_empty_html = ''
```

```
default_error_messages = {}
```

```
initial = ''
```

```
run_validation(data=<class 'rest_framework.fields.empty'>)
```

Validate a simple representation and return the internal value.

The provided data may be *empty* if no representation was included in the input.

May raise *SkipField* if the field should not be included in the validated data.

```
to_internal_value(data)
```

Transform the *incoming* primitive data into a native value.

to_representation (*value*)
 Transform the *outgoing* native value into primitive data.

38.4.1.1.1.267 fobi.contrib.apps.drf_integration.helpers module

`fobi.contrib.apps.drf_integration.helpers.map_field_name_to_label(serializer)`
 Takes a form and creates label to field name map.

Parameters **serializer** – Instance of `rest_framework.serializers.Serializer`.

Return dict

38.4.1.1.1.268 fobi.contrib.apps.drf_integration.metadata module

class `fobi.contrib.apps.drf_integration.metadata.FobiMetaData`

Bases: `rest_framework.metadata.SimpleMetadata`

Meta data for better representation of the form elements.

get_field_info (*field*)

Get field info.

Given an instance of a serializer field, return a dictionary of metadata about it.

label_lookup = `<rest_framework.utils.field_mapping.ClassLookupDict object>`

38.4.1.1.1.269 fobi.contrib.apps.drf_integration.serializers module

class `fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer` (*instance=None*,
data=<class
'rest_framework.fields.empty'
***kwargs*)

Bases: `rest_framework.serializers.ModelSerializer`

FormEntry serializer (read-only).

Used for detail and list views.

class **Meta**

Bases: `object`

Options.

fields = ('url', 'id', 'slug', 'title')

model

alias of `fobi.models.FormEntry`

read_only_fields = ('url', 'id', 'slug', 'title')

get_fields_metadata (*field_name=None*)

Just to make sure nothing breaks.

get_url (*obj*)

Get URL.

38.4.1.1.1.270 fobi.contrib.apps.drf_integration.urls module

38.4.1.1.1.271 fobi.contrib.apps.drf_integration.utils module

`fobi.contrib.apps.drf_integration.utils.get_serializer_class` (*form_entry*, *request=None*, *has_value=None*, *declared_fields=None*)

Get assembled serializer class.

Parameters

- **form_entry** (`fobi.models.FormEntry`) –
- **request** (`django.http.HttpRequest`) –
- **has_value** (*bool*) –
- **declared_fields** (*list*) –

Return `django.forms.Form`

38.4.1.1.1.272 fobi.contrib.apps.drf_integration.views module

class `fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet` (***kwargs*)
 Bases: `rest_framework.mixins.RetrieveModelMixin`, `rest_framework.mixins.UpdateModelMixin`, `rest_framework.mixins.ListModelMixin`, `rest_framework.viewsets.GenericViewSet`

FormEntry view set.

get_object ()
 Override `get_object` to get things done.

get_queryset ()
 Get queryset.

We show all forms to authenticated users and show only public forms to non-authenticated users.

get_serializer (**args*, ***kwargs*)
 Get the serializer.

get_serializer_class ()
 Get serializer class.

has_value ()

lookup_field = 'slug'

lookup_url_kwarg = 'slug'

metadata_class
 alias of `fobi.contrib.apps.drf_integration.metadata.FobiMetaData`

permission_classes = [`<class 'rest_framework.permissions.AllowAny'>`]

queryset

suffix = None

update (*request*, **args*, ***kwargs*)

38.4.1.1.1.273 Module contents

38.4.1.1.1.274 fobi.contrib.apps.feincms_integration package

38.4.1.1.1.275 Submodules

38.4.1.1.1.276 fobi.contrib.apps.feincms_integration.apps module

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

38.4.1.1.1.277 fobi.contrib.apps.feincms_integration.conf module

`fobi.contrib.apps.feincms_integration.conf.get_setting(setting, override=None)`
 Get setting.

Get a setting from `fobi.contrib.apps.feincms_integration.conf` module, falling back to the default.

If `override` is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

38.4.1.1.1.278 fobi.contrib.apps.feincms_integration.defaults module

38.4.1.1.1.279 fobi.contrib.apps.feincms_integration.helpers module

`fobi.contrib.apps.feincms_integration.helpers.get_form_template_choices()`
 Gets the form template choices.

It's possible to provide theme templates per theme or just per project.

Return list

`fobi.contrib.apps.feincms_integration.helpers.get_success_page_template_choices()`
 Get success page template choices.

Return list

38.4.1.1.1.280 fobi.contrib.apps.feincms_integration.settings module

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

38.4.1.1.1.281 fobi.contrib.apps.feincms_integration.widgets module

```
class fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget (*args,
                                                                    **kwargs)
    Bases: django.db.models.base.Model, fobi.integration.processors.
    IntegrationProcessor
```

Widget for to FeinCMS.

Property `fobi.models.FormEntry form_entry` Form entry to be rendered.

Property `str template` If given used for rendering the form.

```
class Meta
    Bases: object
    Meta options.
    abstract = False
    app_label = 'fobi'
```

```
can_redirect = True
```

```
finalize (request, response)
    Finalize.
```

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

In the example:

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

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

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

```
form_sent_get_param = 'sent'
```

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

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

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

```
get_form_template_name_display (**morekwargs)
```

```
get_success_page_template_name_display (**morekwargs)
```

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

hide_success_page_title

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

process (*request*, ***kwargs*)

This is where most of the form handling happens.

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

Return *django.http.HttpResponse* | *str*

render (***kwargs*)

Render.

success_page_template_name

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

success_page_text

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

success_page_title

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

38.4.1.1.1.282 Module contents

38.4.1.1.1.283 fobi.contrib.apps.mezzanine_integration package

38.4.1.1.1.284 Submodules

38.4.1.1.1.285 fobi.contrib.apps.mezzanine_integration.admin module

38.4.1.1.1.286 fobi.contrib.apps.mezzanine_integration.apps module

class *fobi.contrib.apps.mezzanine_integration.apps.Config* (*app_name*,
app_module)

Bases: *django.apps.config AppConfig*

Config.

label = *'fobi_contrib_apps_mezzanine_integration'*

name = *'fobi.contrib.apps.mezzanine_integration'*

38.4.1.1.1.287 fobi.contrib.apps.mezzanine_integration.conf module

fobi.contrib.apps.mezzanine_integration.conf.get_setting (*setting*, *override=None*)

Get setting.

Get a setting from *fobi.contrib.apps.mezzanine_integration* conf module, falling back to the default.

If *override* is not *None*, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

38.4.1.1.1.288 fobi.contrib.apps.mezzanine_integration.defaults module

38.4.1.1.1.289 fobi.contrib.apps.mezzanine_integration.helpers module

`fobi.contrib.apps.mezzanine_integration.helpers.get_form_template_choices()`

Gets the form template choices.

It's possible to provide theme templates per theme or just per project.

Return list

`fobi.contrib.apps.mezzanine_integration.helpers.get_success_page_template_choices()`

Get success page template choices.

Return list

38.4.1.1.1.290 fobi.contrib.apps.mezzanine_integration.models module

38.4.1.1.1.291 fobi.contrib.apps.mezzanine_integration.page_processors module

38.4.1.1.1.292 fobi.contrib.apps.mezzanine_integration.settings module

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

38.4.1.1.1.293 Module contents

38.4.1.1.1.294 Module contents

38.4.1.1.1.295 fobi.contrib.plugins package

38.4.1.1.1.296 Subpackages

38.4.1.1.1.297 fobi.contrib.plugins.form_elements package

38.4.1.1.1.298 Subpackages

38.4.1.1.1.299 fobi.contrib.plugins.form_elements.content package

38.4.1.1.1.300 Subpackages

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

38.4.1.1.1.302 Submodules

38.4.1.1.1.303 fobi.contrib.plugins.form_elements.content.content_image.apps module

```
class fobi.contrib.plugins.form_elements.content.content_image.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_content_content_image'
    name = 'fobi.contrib.plugins.form_elements.content.content_image'
```

38.4.1.1.1.304 fobi.contrib.plugins.form_elements.content.content_image.base module

```
class fobi.contrib.plugins.form_elements.content.content_image.base.ContentImagePlugin(user=
    Bases: fobi.base.FormElementPlugin
    Content image plugin.
    clone_plugin_data(entry)
        Clone plugin data.
        Clone plugin data, which means we make a copy of the original image.
        TODO: Perhaps rely more on data of form_element_entry?
    delete_plugin_data()
        Delete uploaded file.
    form
        alias of fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm
```

```

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

get_raw_data ()
    Get raw data.

    Might be used in integration plugins.

get_rendered_image ()
    Get rendered image.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>

post_processor ()
    Post process data.

    Always the same.

uid = 'content_image'

```

38.4.1.1.1.305 fobi.contrib.plugins.form_elements.content.content_image.conf module

```

fobi.contrib.plugins.form_elements.content.content_image.conf.get_setting (setting,
                                                                           over-
                                                                           ride=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.

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

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

```

class fobi.contrib.plugins.form_elements.content.content_image.fobi_form_elements.ContentImagePlugin
    Bases: fobi.base.FormElementPlugin

    Content image plugin.

    clone_plugin_data (entry)
        Clone plugin data.

        Clone plugin data, which means we make a copy of the original image.

        TODO: Perhaps rely more on data of form_element_entry?

    delete_plugin_data ()
        Delete uploaded file.

```

```

form
    alias of fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None, **kwargs)
    Get form field instances.

get_raw_data ()
    Get raw data.

    Might be used in integration plugins.

get_rendered_image ()
    Get rendered image.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>

post_processor ()
    Post process data.

    Always the same.

uid = 'content_image'

```

38.4.1.1.1.308 fobi.contrib.plugins.form_elements.content.content_image.forms module

```

class fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm (data=..., files=..., auto_i=..., pre=..., fix=No=..., ini=..., tial=N=..., er=..., ror_cl=..., 'djang=..., la=..., bel_su=..., empty=..., field_o=..., use_re=...)

Bases: django.forms.forms.Form, fobi.base.BasePluginForm
Form for ContentImagePlugin.

base_fields = {'alt': <django.forms.fields.CharField object>, 'file': <django.forms.
declared_fields = {'alt': <django.forms.fields.CharField object>, 'file': <django.fo
media
plugin_data_fields = [('file', ''), ('alt', ''), ('fit_method', 'center'), ('size', '5
save_plugin_data (request=None)
    Saving the plugin data and moving the file.

```

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

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

38.4.1.1.1.310 fobi.contrib.plugins.form_elements.content.content_image.settings module

- `FIT_METHOD_CROP_SMART` (string)
- `FIT_METHOD_CROP_CENTER` (string)
- `FIT_METHOD_CROP_SCALE` (string)
- `FIT_METHOD_FIT_WIDTH` (string)
- `FIT_METHOD_FIT_HEIGHT` (string)
- `DEFAULT_FIT_METHOD` (string)
- `FIT_METHODS_CHOICES` (tuple)
- `FIT_METHODS_CHOICES_WITH_EMPTY_OPTION` (list)
- `IMAGES_UPLOAD_DIR` (string)

38.4.1.1.1.311 Module contents

38.4.1.1.1.312 fobi.contrib.plugins.form_elements.content.content_image_url package

38.4.1.1.1.313 Submodules

38.4.1.1.1.314 fobi.contrib.plugins.form_elements.content.content_image_url.apps module

```
class fobi.contrib.plugins.form_elements.content.content_image_url.apps.Config(app_name,  
                                     app_module)  
    Bases: django.apps.config AppConfig  
    Config.  
    label = 'fobi_contrib_plugins_form_elements_content_content_image_url'  
    name = 'fobi.contrib.plugins.form_elements.content.content_image_url'
```

38.4.1.1.1.315 fobi.contrib.plugins.form_elements.content.content_image_url.base module

```
class fobi.contrib.plugins.form_elements.content.content_image_url.base.ContentImageURLPlugin  
    Bases: fobi.base.FormElementPlugin  
    Content image plugin.  
    form  
        alias of fobi.contrib.plugins.form_elements.content.content_image_url.  
        forms.ContentImageURLForm  
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,  
                             **kwargs)  
        Get form field instances.
```

```

get_raw_data()
    Get raw data.

    Might be used in integration plugins.

get_rendered_image()
    Get rendered image.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>

post_processor()
    Post process data.

    Always the same.

uid = 'content_image_url'

```

38.4.1.1.1.316 fobi.contrib.plugins.form_elements.content.content_image_url.conf module

```

fobi.contrib.plugins.form_elements.content.content_image_url.conf.get_setting(setting,
                                                                              over-
                                                                              ride=None)

```

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.content.content_image_url.conf` module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

38.4.1.1.1.317 fobi.contrib.plugins.form_elements.content.content_image_url.defaults module

38.4.1.1.1.318 fobi.contrib.plugins.form_elements.content.content_image_url.fobi_form_elements module

```

class fobi.contrib.plugins.form_elements.content.content_image_url.fobi_form_elements.ContentImageURLForm(
    Bases: fobi.base.FormElementPlugin
)

```

Content image plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_image_url.forms.ContentImageURLForm`

```

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                        **kwargs)

```

Get form field instances.

```

get_raw_data()

```

Get raw data.

Might be used in integration plugins.

```

get_rendered_image()
    Get rendered image.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
post_processor()
    Post process data.

    Always the same.

uid = 'content_image_url'

```

38.4.1.1.1.319 fobi.contrib.plugins.form_elements.content.content_image_url.forms module

```
class fobi.contrib.plugins.form_elements.content.content_image_url.forms.ContentImageURLForm
```

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

Form for ContentImageURLPlugin.

```

base_fields = {'alt': <django.forms.fields.CharField object>, 'fit_method': <django.
declared_fields = {'alt': <django.forms.fields.CharField object>, 'fit_method': <dja
media
plugin_data_fields = [('url', ''), ('alt', ''), ('fit_method', 'fit_width'), ('size',

```

38.4.1.1.1.320 fobi.contrib.plugins.form_elements.content.content_image_url.settings module

- `DEFAULT_SIZE` (string)
- `SIZE_100x100` (string)
- `SIZE_200x200` (string)
- `SIZE_300x300` (string)
- `SIZE_400x400` (string)
- `SIZE_500x500` (string)
- `SIZE_600x600` (tuple)
- `SIZES` (list)

38.4.1.1.1.321 Module contents

38.4.1.1.1.322 fobi.contrib.plugins.form_elements.content.content_markdown package

38.4.1.1.1.323 Submodules

38.4.1.1.1.324 fobi.contrib.plugins.form_elements.content.content_markdown.apps module

```
class fobi.contrib.plugins.form_elements.content.content_markdown.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_content_content_markdown'
    name = 'fobi.contrib.plugins.form_elements.content.content_markdown'
```

38.4.1.1.1.325 fobi.contrib.plugins.form_elements.content.content_markdown.base module

```
class fobi.contrib.plugins.form_elements.content.content_markdown.base.ContentMarkdownPlugin:
    Bases: fobi.base.FormElementPlugin
    Content markdown plugin.

    form
        alias of fobi.contrib.plugins.form_elements.content.content_markdown.
        forms.ContentMarkdownForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get the instances of form fields, that plugin contains.
```

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_entry** (*fobi.models.FormEntry*) –
- **form_element_entries** (*django.db.models.QuerySet*) – Queryset of *fobi.models.FormElementEntry* instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

```
get_raw_data()
```

```
get_rendered_text()
```

Get rendered text.

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

```
html_classes = ['content-markdown']
```

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


post_processor()
 Post-processor (self).
 Redefine in your subclassed plugin when necessary.
 Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.
 Note, that request (django.http.HttpRequest) is available (self.request).

uid = 'content_markdown'

38.4.1.1.1.326 fobi.contrib.plugins.form_elements.content.content_markdown.fobi_form_elements module

class fobi.contrib.plugins.form_elements.content.content_markdown.fobi_form_elements.**ContentMarkdownFormElementPlugin**
 Bases: *fobi.base.FormElementPlugin*

Content markdown plugin.

form
 alias of *fobi.contrib.plugins.form_elements.content.content_markdown.forms.ContentMarkdownForm*

get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
 Get the instances of form fields, that plugin contains.

Parameters

- **request** (django.http.HttpRequest) –
- **form_entry** (fobi.models.FormEntry) –
- **form_element_entries** (django.db.models.QuerySet) – Queryset of *fobi.models.FormElementEntry* instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_raw_data()

get_rendered_text()
 Get rendered text.

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

html_classes = ['content-markdown']

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

post_processor()
 Post-processor (self).

Redefine in your subclassed plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (django.http.HttpRequest) is available (self.request).

```
uid = 'content_markdown'
```

38.4.1.1.1.327 fobi.contrib.plugins.form_elements.content.content_markdown.forms module

```
class fobi.contrib.plugins.form_elements.content.content_markdown.forms.ContentMarkdownForm
```

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

ContentMarkDownForm.

```
base_fields = {'text': <django.forms.fields.CharField object>}
```

```
declared_fields = {'text': <django.forms.fields.CharField object>}
```

```
media
```

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

38.4.1.1.1.328 fobi.contrib.plugins.form_elements.content.content_markdown.widgets module

```
class fobi.contrib.plugins.form_elements.content.content_markdown.widgets.BaseContentMarkd
```

Bases: *fobi.base.FormElementPluginWidget*

Base content markdown form element plugin widget.

```
html_classes = ['content-markdown']
```

```
plugin_uid = 'content_markdown'
```

38.4.1.1.1.329 Module contents

38.4.1.1.1.330 fobi.contrib.plugins.form_elements.content.content_richtext package

38.4.1.1.1.331 Submodules

38.4.1.1.1.332 fobi.contrib.plugins.form_elements.content.content_richtext.apps module

```
class fobi.contrib.plugins.form_elements.content.content_richtext.apps.Config(app_name,  
                                                                           app_module)
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_content_content_richtext'
name = 'fobi.contrib.plugins.form_elements.content.content_richtext'
```

38.4.1.1.1.333 fobi.contrib.plugins.form_elements.content.content_richtext.base module

class fobi.contrib.plugins.form_elements.content.content_richtext.base.ContentRichTextPlugin
Bases: *fobi.base.FormElementPlugin*

Content rich text plugin.

form

alias of *fobi.contrib.plugins.form_elements.content.content_richtext.forms.ContentRichTextForm*

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get the instances of form fields, that plugin contains.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_entry** (*fobi.models.FormEntry*) –
- **form_element_entries** (*django.db.models.QuerySet*) – Queryset of *fobi.models.FormElementEntry* instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_raw_data ()

get_rendered_text ()

Get rendered text.

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

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

post_processor ()

Post-processor (self).

Redefine in your subclassed plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (*django.http.HttpRequest*) is available (*self.request*).

uid = 'content_richtext'

38.4.1.1.1.334 fobi.contrib.plugins.form_elements.content.content_richtext.fobi_form_elements module

class fobi.contrib.plugins.form_elements.content.content_richtext.fobi_form_elements.ContentRichTextForm
Bases: *fobi.base.FormElementPlugin*

Content rich text plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_richtext.forms.ContentRichTextForm`

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get the instances of form fields, that plugin contains.

Parameters

- **request** (`django.http.HttpRequest`) –
- **form_entry** (`fobi.models.FormEntry`) –
- **form_element_entries** (`django.db.models.QuerySet`) – Queryset of `fobi.models.FormElementEntry` instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_raw_data ()

get_rendered_text ()

Get rendered text.

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

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

post_processor ()

Post-processor (self).

Redefine in your subclassed plugin when necessary.

Post process plugin data here (before rendering). This method is being called after the data has been loaded into the plugin.

Note, that request (`django.http.HttpRequest`) is available (`self.request`).

uid = 'content_richtext'

38.4.1.1.1.335 fobi.contrib.plugins.form_elements.content.content_richtext.forms module

class fobi.contrib.plugins.form_elements.content.content_richtext.forms.**ContentRichTextForm**

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

ContentRichTextForm.

base_fields = {'text': <django.forms.fields.CharField object>}

clean_text()

declared_fields = {'text': <django.forms.fields.CharField object>}

media

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

38.4.1.1.1.336 fobi.contrib.plugins.form_elements.content.content_richtext.widgets module

class fobi.contrib.plugins.form_elements.content.content_richtext.widgets.**BaseContentRichTextFormWidget**

Bases: *fobi.base.FormElementPluginWidget*

Base content rich text form element plugin widget.

html_classes = ['content-richtext']

media_js = ['ckeditor/ckeditor-init.js', 'ckeditor/ckeditor/ckeditor.js']

plugin_uid = 'content_richtext'

38.4.1.1.1.337 Module contents

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

38.4.1.1.1.339 Submodules

38.4.1.1.1.340 fobi.contrib.plugins.form_elements.content.content_text.apps module

class fobi.contrib.plugins.form_elements.content.content_text.apps.**Config**(*app_name*,
app_module)

Bases: django.apps.config.AppConfig

Config.

```
label = 'fobi_contrib_plugins_form_elements_content_content_text'
name = 'fobi.contrib.plugins.form_elements.content.content_text'
```

38.4.1.1.1.341 fobi.contrib.plugins.form_elements.content.content_text.base module

```
class fobi.contrib.plugins.form_elements.content.content_text.base.ContentTextPlugin(user=None)
    Bases: fobi.base.FormElementPlugin
    Content text plugin.

    form
        alias of fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

    get_raw_data()
        Get raw data.
        Might be used in integration plugins.

    get_rendered_text()
        Get rendered text.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    post_processor()
        Post process data.
        Always the same.

    uid = 'content_text'
```

38.4.1.1.1.342 fobi.contrib.plugins.form_elements.content.content_text.conf module

```
fobi.contrib.plugins.form_elements.content.content_text.conf.get_setting(setting, override=None)

    Get setting.

    Get a setting from fobi.contrib.plugins.form_elements.content.content_text conf
    module, falling back to the default.

    If override is not None, it will be used instead of the setting.
```

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

38.4.1.1.1.343 fobi.contrib.plugins.form_elements.content.content_text.defaults module

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

```
class fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextFormElement
    Bases: fobi.base.FormElementPlugin
    Content text plugin.

    form
        alias of fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    get_raw_data()
        Get raw data.

        Might be used in integration plugins.

    get_rendered_text()
        Get rendered text.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    post_processor()
        Post process data.

        Always the same.

    uid = 'content_text'
```

38.4.1.1.1.345 fobi.contrib.plugins.form_elements.content.content_text.forms module

```
class fobi.contrib.plugins.form_elements.content.content_text.forms.ContentTextForm(data=None, files=None,
auto_id=None, pre_fix=None, initial=None, error_class=None, 'django.forms.labeled_suffix', empty_password_field_order, use_required_attribute)
    Bases: django.forms.forms.Form, fobi.base.BasePluginForm
    Form for ContentTextPlugin.

    base_fields = {'text': <django.forms.fields.CharField object>}
```

```

clean_text ()
    Clean text value.

declared_fields = {'text': <django.forms.fields.CharField object>}

media

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

38.4.1.1.1.346 fobi.contrib.plugins.form_elements.content.content_text.settings module

38.4.1.1.1.347 Module contents

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

38.4.1.1.1.349 Submodules

38.4.1.1.1.350 fobi.contrib.plugins.form_elements.content.content_video.apps module

```

class fobi.contrib.plugins.form_elements.content.content_video.apps.Config(app_name,
                                                                           app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_content_content_video'
    name = 'fobi.contrib.plugins.form_elements.content.content_video'
```

38.4.1.1.1.351 fobi.contrib.plugins.form_elements.content.content_video.base module

```

class fobi.contrib.plugins.form_elements.content.content_video.base.ContentVideoPlugin(user=
    Bases: fobi.base.FormElementPlugin
    Content video plugin.

    form
        alias of fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    get_raw_data ()
        Get raw data.

        Might be used in integration plugins.

    get_rendered_video ()
        Get rendered video.

        Might be used in integration plugins.

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



```

post_processor()
    Process plugin data.

    Always the same.

uid = 'content_video'

```

38.4.1.1.1.352 fobi.contrib.plugins.form_elements.content.content_video.conf module

```

fobi.contrib.plugins.form_elements.content.content_video.conf.get_setting(setting,
                                                                           over-
                                                                           ride=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.

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

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

```

class fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.ContentV
Bases: fobi.base.FormElementPlugin

```

Content video plugin.

form

alias of `fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm`

```

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                        **kwargs)

```

Get form field instances.

```

get_raw_data()

```

Get raw data.

Might be used in integration plugins.

```

get_rendered_video()

```

Get rendered video.

Might be used in integration plugins.

```

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

```

```

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

```

```

post_processor()
    Process plugin data.

    Always the same.

uid = 'content_video'

```

38.4.1.1.1.355 fobi.contrib.plugins.form_elements.content.content_video.forms module

```

class fobi.contrib.plugins.form_elements.content.content_video.forms.ContentVideoForm(data=
files=N
auto_i
pre-
fix=No
ini-
tial=N
er-
ror_cl
'djang
la-
bel_su
empty_
field_o
use_re

```

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

Form for ContentVideoPlugin.

```

base_fields = {'size': <django.forms.fields.ChoiceField object>, 'title': <django.f
declared_fields = {'size': <django.forms.fields.ChoiceField object>, 'title': <djanga
media
plugin_data_fields = [('title', ''), ('url', ''), ('size', '500x400')]

```

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

38.4.1.1.1.357 Module contents

38.4.1.1.1.358 Module contents

38.4.1.1.1.359 fobi.contrib.plugins.form_elements.fields package

38.4.1.1.1.360 Subpackages

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

38.4.1.1.1.362 Submodules

38.4.1.1.1.363 fobi.contrib.plugins.form_elements.fields.boolean.apps module

```
class fobi.contrib.plugins.form_elements.fields.boolean.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_boolean'
    name = 'fobi.contrib.plugins.form_elements.fields.boolean'
```

38.4.1.1.1.364 fobi.contrib.plugins.form_elements.fields.boolean.base module

```
class fobi.contrib.plugins.form_elements.fields.boolean.base.BooleanSelectPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Boolean select plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.boolean.forms.
                           BooleanSelectForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'boolean'
```

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

```
class fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements.BooleanSelectPl
    Bases: fobi.base.FormFieldPlugin
    Boolean select plugin.
```

```

form
    alias      of      fobi.contrib.plugins.form_elements.fields.boolean.forms.
                        BooleanSelectForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                        **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'boolean'

```

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

```

class fobi.contrib.plugins.form_elements.fields.boolean.forms.BooleanSelectForm(data=None,
                                     files=None,
                                     auto_id='id_%s',
                                     pre-
                                     fix=None,
                                     ini-
                                     tial=None,
                                     er-
                                     ror_class=<django.forms.utils.error_handler_late_renderer_class object>,
                                     label_suffix=None,
                                     empty_permitted=False,
                                     field_order=None,
                                     use_required_attribute=True)

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

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.BooleanField object>}
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.BooleanField object>}
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('required', '')]

```

38.4.1.1.1.367 Module contents

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

38.4.1.1.1.369 Submodules

38.4.1.1.1.370 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps module

```

class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps.Config(app_name='fobi',
                                     app_model='fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple',
                                     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'
```

38.4.1.1.1.371 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.base module

class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.base.CheckboxSelectMultiple

Bases: *fobi.base.FormFieldPlugin*

Checkbox select multiple field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleInputForm*

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'checkbox_select_multiple'

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

fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf.**get_setting**(*setting, override=None*)

Get setting.

Get a setting from *fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name

- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

38.4.1.1.1.374 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements.

Bases: *fobi.base.FormFieldPlugin*

Checkbox select multiple field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.forms.CheckboxSelectMultipleInputForm*

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'checkbox_select_multiple'

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

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

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

Form for CheckboxSelectMultipleInputPlugin.

base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <djan

clean_initial()

Validating the initial value.

declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <

media

plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''),

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

38.4.1.1.1.377 Module contents

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

38.4.1.1.1.379 Submodules

38.4.1.1.1.380 fobi.contrib.plugins.form_elements.fields.date.apps module

class fobi.contrib.plugins.form_elements.fields.date.apps.**Config**(*app_name*,
app_module)

Bases: django.apps.config AppConfig

Config.

label = 'fobi_contrib_plugins_form_elements_fields_date'

name = 'fobi.contrib.plugins.form_elements.fields.date'

38.4.1.1.1.381 fobi.contrib.plugins.form_elements.fields.date.base module

```
class fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin(user=None)  
    Bases: fobi.base.FormFieldPlugin
```

Date field plugin.

```
form  
    alias      of      fobi.contrib.plugins.form_elements.fields.date.forms.  
                        DateInputForm
```

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None,  
                          **kwargs)
```

Get form field instances.

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

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

```
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)  
    Submit plugin form data/process.
```

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

```
uid = 'date'
```

38.4.1.1.1.382 fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements.DateInputPlugin(use  
    Bases: fobi.base.FormFieldPlugin
```

Date field plugin.

```
form  
    alias      of      fobi.contrib.plugins.form_elements.fields.date.forms.  
                        DateInputForm
```

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None,  
                          **kwargs)
```

Get form field instances.

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

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

```
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)  
    Submit plugin form data/process.
```

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –


```
uid = 'date'
```

38.4.1.1.1.383 fobi.contrib.plugins.form_elements.fields.date.forms module

```
class fobi.contrib.plugins.form_elements.fields.date.forms.DateInputForm (data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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 DateInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan
clean_initial()
    Clean the initial value.
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''),
```

38.4.1.1.1.384 fobi.contrib.plugins.form_elements.fields.date.widgets module

```
class fobi.contrib.plugins.form_elements.fields.date.widgets.BaseDatePluginWidget (plugin)
    Bases: fobi.base.FormElementPluginWidget
    Base date form element plugin widget.
    plugin_uid = 'date'
```

38.4.1.1.1.385 Module contents

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

38.4.1.1.1.387 Submodules

38.4.1.1.1.388 fobi.contrib.plugins.form_elements.fields.date_drop_down.apps module

```
class fobi.contrib.plugins.form_elements.fields.date_drop_down.apps.Config (app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
```

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_date_drop_down'
name = 'fobi.contrib.plugins.form_elements.fields.date_drop_down'
```

38.4.1.1.1.389 fobi.contrib.plugins.form_elements.fields.date_drop_down.base module

```
class fobi.contrib.plugins.form_elements.fields.date_drop_down.base.DateDropDownInputPlugin
    Bases: fobi.base.FormFieldPlugin
```

Date drop down field plugin.

```
form
    alias of fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.
    DateDropDownInputForm
```

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
    **kwargs)
```

Get form field instances.

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

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

```
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

```
uid = 'date_drop_down'
```

38.4.1.1.1.390 fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_elements.DateDropDownInputPlugin
    Bases: fobi.base.FormFieldPlugin
```

Date drop down field plugin.

```
form
    alias of fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.
    DateDropDownInputForm
```

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
    **kwargs)
```

Get form field instances.

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

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

```
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

`uid = 'date_drop_down'`

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

class `fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateDropDownInputForm`

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

Form for `DateDropDownInputPlugin`.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

media

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('year_min', '')]

38.4.1.1.1.392 Module contents

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

38.4.1.1.1.394 Submodules

38.4.1.1.1.395 `fobi.contrib.plugins.form_elements.fields.datetime.apps` module

class `fobi.contrib.plugins.form_elements.fields.datetime.apps.Config`(*app_name*, *app_module*)

Bases: `django.apps.config AppConfig`

Config.

label = 'fobi_contrib_plugins_form_elements_fields_datetime'

name = 'fobi.contrib.plugins.form_elements.fields.datetime'

38.4.1.1.1.396 fobi.contrib.plugins.form_elements.fields.datetime.base module

class fobi.contrib.plugins.form_elements.fields.datetime.base.**DateTimeInputPlugin** (*user=None*)

Bases: *fobi.base.FormFieldPlugin*

DateTime field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm*

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'datetime'

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

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

Bases: *fobi.base.FormFieldPlugin*

DateTime field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeInputForm*

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

```
uid = 'datetime'
```

38.4.1.1.1.398 fobi.contrib.plugins.form_elements.fields.datetime.forms module

```
class fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldInputForm (data=None,
files=None,
auto_id='id_%d',
pre-
fix=None,
ini-
tial=None,
er-
ror_class=<django.forms.widgets.TextInput class>,
'django.forms.widgets.TextInput class',
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 = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.widgets.TextInput class>}
clean_initial()
    Clean the initial value.

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.widgets.TextInput class>}
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('required', 'required')]
```

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

```
class fobi.contrib.plugins.form_elements.fields.datetime.widgets.BaseDateTimePluginWidget (fobi.base.BaseFormElementPluginWidget)
Bases: fobi.base.BaseFormElementPluginWidget
Base datetime form element plugin widget.

plugin_uid = 'datetime'
```

38.4.1.1.1.400 Module contents

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

38.4.1.1.1.402 Submodules

38.4.1.1.1.403 fobi.contrib.plugins.form_elements.fields.decimal.apps module

```
class fobi.contrib.plugins.form_elements.fields.decimal.apps.Config (app_name, app_module)
Bases: django.apps.config.AppConfig
```

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_decimal'
name = 'fobi.contrib.plugins.form_elements.fields.decimal'
```

38.4.1.1.1.404 fobi.contrib.plugins.form_elements.fields.decimal.base module

```
class fobi.contrib.plugins.form_elements.fields.decimal.base.DecimalInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
```

Decimal input plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.decimal.forms.
                        DecimalInputForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                          **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'decimal'
```

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

```
class fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_elements.DecimalInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
```

Decimal input plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.decimal.forms.
                        DecimalInputForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                          **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'decimal'
```

38.4.1.1.1.406 fobi.contrib.plugins.form_elements.fields.decimal.forms module

```
class fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalInputForm (data=None,
                                     files=None,
                                     auto_id='id_%s',
                                     pre-
                                     fix=None,
                                     ini-
                                     tial=None,
                                     er-
                                     ror_class=<class 'django.forms.util.ErrorList'>,
                                     label_suffix=None,
                                     empty_permitted=False,
                                     field_order=None,
                                     use_required_attribute=True)

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

base_fields = {'decimal_places': <django.forms.fields.IntegerField object>, 'help_text': ''}

clean()
    Validating the values.

declared_fields = {'decimal_places': <django.forms.fields.IntegerField object>, 'help_text': ''}

media
    Media for this form.

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

quantize (value, decimal_places, max_digits)
    Quantize the decimal value to the configured precision.
```

38.4.1.1.1.407 Module contents

38.4.1.1.1.408 fobi.contrib.plugins.form_elements.fields.duration package

38.4.1.1.1.409 Submodules

38.4.1.1.1.410 fobi.contrib.plugins.form_elements.fields.duration.apps module

```
class fobi.contrib.plugins.form_elements.fields.duration.apps.Config (app_name,
                                                                    app_module)

Bases: django.apps.config.AppConfig
Config.

label = 'fobi_contrib_plugins_form_elements_fields_duration'

name = 'fobi.contrib.plugins.form_elements.fields.duration'
```

38.4.1.1.1.411 fobi.contrib.plugins.form_elements.fields.duration.base module

```
class fobi.contrib.plugins.form_elements.fields.duration.base.DurationInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin

    Duration field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.duration.forms.
                           DurationInputForm

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                              **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    prepare_plugin_form_data (cleaned_data)
        Prepare plugin form data.
        Might be used in integration plugins.

    submit_plugin_form_data (form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.
          FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'duration'
```

38.4.1.1.1.412 fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.duration.fobi_form_elements.DurationInputP
    Bases: fobi.base.FormFieldPlugin

    Duration field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.duration.forms.
                           DurationInputForm

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                              **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    prepare_plugin_form_data (cleaned_data)
        Prepare plugin form data.
        Might be used in integration plugins.
```


submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'duration'

38.4.1.1.1.413 fobi.contrib.plugins.form_elements.fields.duration.forms module

```
class fobi.contrib.plugins.form_elements.fields.duration.forms.DurationInputForm (data=None, files=None, auto_id='id_%d', pre-fix=None, initial=None, error_class=<django.forms.models.BaseForm>, label_suffix=None, empty_permitted=False, field_order=None, use_required_attribute=True)
```

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

Form for DurationInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}
clean()
    Validation.
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}
media
```

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('required', '')]
render_plugin_data_fields = <django.forms.render_plugin_data_fields object>
```

38.4.1.1.1.414 Module contents

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

38.4.1.1.1.416 Submodules

38.4.1.1.1.417 fobi.contrib.plugins.form_elements.fields.email.apps module

```
class fobi.contrib.plugins.form_elements.fields.email.apps.Config (app_name, app_module)
```

Bases: *django.apps.config.AppConfig*

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_email'
name = 'fobi.contrib.plugins.form_elements.fields.email'
```

38.4.1.1.1.418 fobi.contrib.plugins.form_elements.fields.email.base module

```
class fobi.contrib.plugins.form_elements.fields.email.base.EmailInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
```

Email input plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.email.forms.
                        EmailInputForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                        **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'email'
```

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

```
class fobi.contrib.plugins.form_elements.fields.email.fobi_form_elements.EmailInputPlugin(
    Bases: fobi.base.FormFieldPlugin
```

Email input plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.email.forms.
                        EmailInputForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                        **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'email'
```

38.4.1.1.1.420 fobi.contrib.plugins.form_elements.fields.email.forms module

```
class fobi.contrib.plugins.form_elements.fields.email.forms.EmailInputForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    pre-
                                                                    fix=None,
                                                                    ini-
                                                                    tial=None,
                                                                    er-
                                                                    ror_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 EmailInputPlugin.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

clean()
    Hook for doing any extra form-wide cleaning after Field.clean() has been called on every field. Any
    ValidationError raised by this method will not be associated with a particular field; it will have a special-
    case association with the field named '__all__'.

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

media
    Media for this form.

plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('required', '')]
    Fields for this plugin.
```

38.4.1.1.1.421 Module contents

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

38.4.1.1.1.423 Submodules

38.4.1.1.1.424 fobi.contrib.plugins.form_elements.fields.file.apps module

```
class fobi.contrib.plugins.form_elements.fields.file.apps.Config(app_name,
                                                                    app_module)

Bases: django.apps.config.AppConfig
Config.

label = 'fobi_contrib_plugins_form_elements_fields_file'
name = 'fobi.contrib.plugins.form_elements.fields.file'
```

38.4.1.1.1.425 fobi.contrib.plugins.form_elements.fields.file.base module

```
class fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin(user=None)

Bases: fobi.base.BaseFormFieldPlugin
```

File field plugin.

form
alias of `fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm`

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

prepare_plugin_form_data (*cleaned_data*)

Prepare plugin form data.

Parameters *cleaned_data* –

Returns

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process file upload.

Handling the posted data for file plugin when form is submitted. This method affects the original form and that's why it returns it.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** –

uid = 'file'

38.4.1.1.1.426 fobi.contrib.plugins.form_elements.fields.file.conf module

`fobi.contrib.plugins.form_elements.fields.file.conf.get_setting` (*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.conf` module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

38.4.1.1.1.428 fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements.FileInputPlugin` (*use*

Bases: `fobi.base.FormFieldPlugin`

File field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm`

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

prepare_plugin_form_data (*cleaned_data*)

Prepare plugin form data.

Parameters *cleaned_data* –

Returns

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process file upload.

Handling the posted data for file plugin when form is submitted. This method affects the original form and that's why it returns it.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –
- **form_element_entries** –

uid = `'file'`

38.4.1.1.1.429 fobi.contrib.plugins.form_elements.fields.file.forms module

```
class fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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 FileInputPlugin.

```
base_fields = {'allowed_extensions': <django.forms.fields.CharField object>, 'help_te
```

```
clean()
```

Hook for doing any extra form-wide cleaning after Field.clean() has been called on every field. Any ValidationError raised by this method will not be associated with a particular field; it will have a special-case association with the field named '__all__'.

```
declared_fields = {'allowed_extensions': <django.forms.fields.CharField object>, 'hel
```

```
media
```

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

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

- FILES_UPLOAD_DIR (string)

38.4.1.1.1.431 Module contents

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

38.4.1.1.1.433 Submodules

38.4.1.1.1.434 fobi.contrib.plugins.form_elements.fields.float.apps module

```
class fobi.contrib.plugins.form_elements.fields.float.apps.Config(app_name,
                                                                    app_module)
```

Bases: django.apps.config.AppConfig

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_float'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.float'
```

38.4.1.1.1.435 fobi.contrib.plugins.form_elements.fields.float.base module

```
class fobi.contrib.plugins.form_elements.fields.float.base.FloatInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin
```

Float input plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.float.forms.
                        FloatInputForm
```

```
get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
```

Get form field instances.

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

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

```
uid = 'float'
```

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

```
class fobi.contrib.plugins.form_elements.fields.float.fobi_form_elements.FloatInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin
```

Float input plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.float.forms.
                        FloatInputForm
```

```
get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
```

Get form field instances.

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

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

```
uid = 'float'
```

38.4.1.1.1.437 fobi.contrib.plugins.form_elements.fields.float.forms module

```
class fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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=
```

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

Form for FloatInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}
```

```
clean()
```

Validating the values.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}
```

```
media
```

```
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('media', '')]
```

38.4.1.1.1.438 Module contents

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

38.4.1.1.1.440 Submodules

38.4.1.1.1.441 fobi.contrib.plugins.form_elements.fields.hidden.apps module

```
class fobi.contrib.plugins.form_elements.fields.hidden.apps.Config(app_name,
                                                                    app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_hidden'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.hidden'
```

38.4.1.1.1.442 fobi.contrib.plugins.form_elements.fields.hidden.base module

```
class fobi.contrib.plugins.form_elements.fields.hidden.base.HiddenInputPlugin(user=None)
```

Bases: `fobi.base.BaseFormFieldPlugin`

Hidden field plugin.


```

form
    alias      of      fobi.contrib.plugins.form_elements.fields.hidden.forms.
                        HiddenInputForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
is_hidden = True
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'hidden'

```

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

```

class fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_elements.HiddenInputPlugin
    Bases: fobi.base.FormFieldPlugin
    Hidden field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.hidden.forms.
                            HiddenInputForm

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    is_hidden = True
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'hidden'

```

38.4.1.1.1.444 fobi.contrib.plugins.form_elements.fields.hidden.forms module

```

class fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm (data=None,
                                                                              files=None,
                                                                              auto_id='id_%s',
                                                                              pre-
                                                                              fix=None,
                                                                              ini-
                                                                              tial=None,
                                                                              er-
                                                                              ror_class=<class
                                                                              'django.forms.utils.L
                                                                              La-
                                                                              bel_suffix=None,
                                                                              empty_permitted=F
                                                                              ield_order=None,
                                                                              use_required_attri
                                                                              b
                                                                              ut
                                                                              es
                                                                              =None
                                                                              )

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

```

```

base_fields = {'initial': <django.forms.fields.CharField object>, 'label': <django.f
clean()
    Hook for doing any extra form-wide cleaning after Field.clean() has been called on every field. Any
    ValidationError raised by this method will not be associated with a particular field; it will have a special-
    case association with the field named '__all__'.

declared_fields = {'initial': <django.forms.fields.CharField object>, 'label': <djan
media
plugin_data_fields = [('label', ''), ('name', ''), ('initial', ''), ('max_length', '25

```

38.4.1.1.1.445 Module contents

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

38.4.1.1.1.447 Submodules

38.4.1.1.1.448 fobi.contrib.plugins.form_elements.fields.input.apps module

```

class fobi.contrib.plugins.form_elements.fields.input.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_input'
    name = 'fobi.contrib.plugins.form_elements.fields.input'

```

38.4.1.1.1.449 fobi.contrib.plugins.form_elements.fields.input.base module

```

class fobi.contrib.plugins.form_elements.fields.input.base.InputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Input field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.input.forms.InputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'input'

```

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

38.4.1.1.1.451 fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements module

```

class fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements.InputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

```

Input field plugin.

```
form
    alias of fobi.contrib.plugins.form_elements.fields.input.forms.InputForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'input'
```

38.4.1.1.1.452 fobi.contrib.plugins.form_elements.fields.input.forms module

```
class fobi.contrib.plugins.form_elements.fields.input.forms.InputForm (data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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 = {'autocomplete_value': <django.forms.fields.BooleanField object>, 'auto-
clean()
    Hook for doing any extra form-wide cleaning after Field.clean() has been called on every field. Any
    ValidationError raised by this method will not be associated with a particular field; it will have a special-
    case association with the field named '__all__'.

declared_fields = {'autocomplete_value': <django.forms.fields.BooleanField object>, '
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''),
```

38.4.1.1.1.453 Module contents

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

38.4.1.1.1.455 Submodules

38.4.1.1.1.456 fobi.contrib.plugins.form_elements.fields.integer.apps module

```
class fobi.contrib.plugins.form_elements.fields.integer.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_integer'
    name = 'fobi.contrib.plugins.form_elements.fields.integer'
```

38.4.1.1.1.457 fobi.contrib.plugins.form_elements.fields.integer.base module

```
class fobi.contrib.plugins.form_elements.fields.integer.base.IntegerInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Integer input plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.integer.forms.
                           IntegerInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'integer'
```

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

```
class fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements.IntegerInputPlu
    Bases: fobi.base.FormFieldPlugin
    Integer input plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.integer.forms.
                           IntegerInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'integer'
```

38.4.1.1.1.459 fobi.contrib.plugins.form_elements.fields.integer.forms module

```
class fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm (data=None,
                                     files=None,
                                     auto_id='id_%s',
                                     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 IntegerInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django
```

```
clean()
```

Validating the values.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
```

```
media
```

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

38.4.1.1.1.460 Module contents

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

38.4.1.1.1.462 Submodules

38.4.1.1.1.463 fobi.contrib.plugins.form_elements.fields.ip_address.apps module

```
class fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config (app_name,
                                                                           app_module)
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_ip_address'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.ip_address'
```

38.4.1.1.1.464 fobi.contrib.plugins.form_elements.fields.ip_address.base module

```
class fobi.contrib.plugins.form_elements.fields.ip_address.base.IPAddressInputPlugin (user=None)
Bases: fobi.base.FormFieldPlugin
```

IP address field plugin.

```

form
    alias of fobi.contrib.plugins.form_elements.fields.ip_address.forms.
    IPAddressInputForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'ip_address'

```

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

```

class fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_elements.IPAddressInputForm
    Bases: fobi.base.FormFieldPlugin
    IP address field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.ip_address.forms.
        IPAddressInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'ip_address'

```

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

```

class fobi.contrib.plugins.form_elements.fields.ip_address.forms.IPAddressInputForm(data=None, files=None, auto_id=None, pre_fix=None, initial=None, error_class=None, 'django.f la- bel_suffix empty_pe field_orde use_requi

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

    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django

```

`clean()`

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

`declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <`

`media`

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

38.4.1.1.1.467 Module contents

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

38.4.1.1.1.469 Submodules

38.4.1.1.1.470 `fobi.contrib.plugins.form_elements.fields.null_boolean.apps` module

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_null_boolean'
    name = 'fobi.contrib.plugins.form_elements.fields.null_boolean'
```

38.4.1.1.1.471 `fobi.contrib.plugins.form_elements.fields.null_boolean.base` module

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.base.NullBooleanSelectPlugin(
    Bases: fobi.base.FormFieldPlugin
    Null boolean select plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.null_boolean.forms.NullBooleanSelectForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                            **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'null_boolean'
```

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

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_elements.NullBoolean
    Bases: fobi.base.FormFieldPlugin
    Null boolean select plugin.
```

```

form
    alias of fobi.contrib.plugins.form_elements.fields.null_boolean.forms.
             NullBooleanSelectForm

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

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

```

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

```

class fobi.contrib.plugins.form_elements.fields.null_boolean.forms.NullBooleanSelectForm(da
    fil
    au
    pr
    fix
    in
    tic
    en
    ro
    'd
    la
    be
    en
    fie
    us

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

    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan
    declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
    media
    plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''),

```

38.4.1.1.1.474 Module contents

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

38.4.1.1.1.476 Submodules

38.4.1.1.1.477 *fobi.contrib.plugins.form_elements.fields.password.apps* module

```

class fobi.contrib.plugins.form_elements.fields.password.apps.Config(app_name,
                                                                    app_module)

    Bases: django.apps.config.AppConfig
    Config.

```



```
label = 'fobi_contrib_plugins_form_elements_fields_password'
name = 'fobi.contrib.plugins.form_elements.fields.password'
```

38.4.1.1.1.478 fobi.contrib.plugins.form_elements.fields.password.base module

```
class fobi.contrib.plugins.form_elements.fields.password.base.PasswordInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin
    Password field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.password.forms.
                             PasswordInputForm

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                               **kwargs)
        Get form field instances.

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

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

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

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.password.forms.
                             PasswordInputForm

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                               **kwargs)
        Get form field instances.

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

38.4.1.1.1.480 fobi.contrib.plugins.form_elements.fields.password.forms module

```
class fobi.contrib.plugins.form_elements.fields.password.forms.PasswordInputForm(data=None,
                                     files=None,
                                     auto_id='id_%d',
                                     pre-
                                     fix=None,
                                     ini-
                                     tial=None,
                                     er-
                                     ror_class=<django.forms
                                     'django.forms
                                     la-
                                     bel_suffix=None,
                                     empty_permit
                                     field_order=None,
                                     use_required_
```

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

Form for PasswordInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django
```

```
clean()
```

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django
```

```
media
```

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

38.4.1.1.1.481 Module contents

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

38.4.1.1.1.483 Submodules

38.4.1.1.1.484 fobi.contrib.plugins.form_elements.fields.radio.apps module

```
class fobi.contrib.plugins.form_elements.fields.radio.apps.Config(app_name,
                                                                app_module)
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_radio'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.radio'
```

38.4.1.1.1.485 fobi.contrib.plugins.form_elements.fields.radio.base module

```
class fobi.contrib.plugins.form_elements.fields.radio.base.RadioInputPlugin(user=None)
```

Bases: `fobi.base.BaseFormFieldPlugin`

Radio field plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.radio.forms.
                        RadioInputForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
submit_plugin_form_data (form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.

Parameters
    • form_entry (fobi.models.FormEntry) – Instance of fobi.models.
      FormEntry.
    • request (django.http.HttpRequest) –
    • form (django.forms.Form) –

uid = 'radio'
```

38.4.1.1.1.486 *fobi.contrib.plugins.form_elements.fields.radio.conf* module

```
fobi.contrib.plugins.form_elements.fields.radio.conf.get_setting (setting, 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.

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

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

```
class fobi.contrib.plugins.form_elements.fields.radio.fobi_form_elements.RadioInputPlugin ()
    Bases: fobi.base.FormFieldPlugin
```

Radio field plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.radio.forms.
                        RadioInputForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.
```

```
group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

```
uid = 'radio'
```

38.4.1.1.1.489 fobi.contrib.plugins.form_elements.fields.radio.forms module

```
class fobi.contrib.plugins.form_elements.fields.radio.forms.RadioInputForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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 `RadioInputPlugin`.

```
base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <django.forms.fields.CharField object>}
clean_initial()
    Validating the initial value.
declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <django.forms.fields.CharField object>}
media
plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''), ('initial', '')]
```

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

38.4.1.1.1.491 Module contents

38.4.1.1.1.492 fobi.contrib.plugins.form_elements.fields.range_select package

38.4.1.1.1.493 Submodules

38.4.1.1.1.494 fobi.contrib.plugins.form_elements.fields.range_select.apps module

```
class fobi.contrib.plugins.form_elements.fields.range_select.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_range_select'
    name = 'fobi.contrib.plugins.form_elements.fields.range_select'
```

38.4.1.1.1.495 fobi.contrib.plugins.form_elements.fields.range_select.base module

```
class fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSelectInputPlugin(us
    Bases: fobi.base.FormFieldPlugin
    Range select input plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.range_select.forms.
        RangeSelectInputForm

    get_choices()
        Get choices.

        Might be used in integration plugins.

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    get_initial()
        Get initial value.

        Might be used in integration plugins.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'range_select'
```

38.4.1.1.1.496 fobi.contrib.plugins.form_elements.fields.range_select.conf module

```
fobi.contrib.plugins.form_elements.fields.range_select.conf.get_setting(setting,
                                                                    over-
                                                                    ride=None)

    Get setting.
```

Get a setting from *fobi.contrib.plugins.form_elements.fields.range_select* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

38.4.1.1.1.497 fobi.contrib.plugins.form_elements.fields.range_select.defaults module

38.4.1.1.1.498 fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements.RangeSelect`

Bases: `fobi.base.FormFieldPlugin`

Range select input plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.range_select.forms.RangeSelectInputForm`

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

get_initial()

Get initial value.

Might be used in integration plugins.

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

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

uid = `'range_select'`

38.4.1.1.1.499 fobi.contrib.plugins.form_elements.fields.range_select.forms module

```
class fobi.contrib.plugins.form_elements.fields.range_select.forms.RangeSelectInputForm(data=
files
auto
pre-
fix=
ini-
tial
er-
ror_
'dja
la-
bel_
emp
field
use.

Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
Form for RangeSelectInputPlugin.
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan
clean()
    Validating the values.
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
media
plugin_data_fields = [('label', ''), ('name', ''), ('min_value', 0), ('max_value', 100
```

38.4.1.1.1.500 fobi.contrib.plugins.form_elements.fields.range_select.settings module

38.4.1.1.1.501 Module contents

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

38.4.1.1.1.503 Submodules

38.4.1.1.1.504 fobi.contrib.plugins.form_elements.fields.regex.apps module

```
class fobi.contrib.plugins.form_elements.fields.regex.apps.Config(app_name,
                                                                    app_module)

Bases: django.apps.config AppConfig
Config.
label = 'fobi_contrib_plugins_form_elements_fields_regex'
name = 'fobi.contrib.plugins.form_elements.fields.regex'
```

38.4.1.1.1.505 fobi.contrib.plugins.form_elements.fields.regex.base module

```
class fobi.contrib.plugins.form_elements.fields.regex.base.RegexInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin

    Regex field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                               **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'regex'
```

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

```
class fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.RegexInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin

    Regex field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                               **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'regex'
```


38.4.1.1.1.507 fobi.contrib.plugins.form_elements.fields.regex.forms module

```
class fobi.contrib.plugins.form_elements.fields.regex.forms.RegexInputForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    pre-
                                                                    fix=None,
                                                                    ini-
                                                                    tial=None,
                                                                    er-
                                                                    ror_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 RegexInputPlugin.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}
clean()
    Validation.

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''), ('errors', '')]
```

38.4.1.1.1.508 Module contents

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

38.4.1.1.1.510 Submodules

38.4.1.1.1.511 fobi.contrib.plugins.form_elements.fields.select.apps module

```
class fobi.contrib.plugins.form_elements.fields.select.apps.Config(app_name,
                                                                    app_module)

Bases: django.apps.config.AppConfig
Config.

label = 'fobi_contrib_plugins_form_elements_fields_select'
name = 'fobi.contrib.plugins.form_elements.fields.select'
```

38.4.1.1.1.512 fobi.contrib.plugins.form_elements.fields.select.base module

```
class fobi.contrib.plugins.form_elements.fields.select.base.SelectInputPlugin(user=None)
Bases: fobi.base.FormFieldPlugin
Select field plugin.
```

form
 alias of `fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm`

get_choices()
 Get choices.
 Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
 Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)
 Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select'

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

`fobi.contrib.plugins.form_elements.fields.select.conf.get_setting`(*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.select` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

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

class `fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.SelectInputPlugin`
 Bases: `fobi.base.FormFieldPlugin`

Select field plugin.

form
 alias of `fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm`

```

get_choices()
    Get choices.

    Might be used in integration plugins.

get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                        **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.

Parameters
    • form_entry (fobi.models.FormEntry) – Instance of fobi.models.
      FormEntry.
    • request (django.http.HttpRequest) –
    • form (django.forms.Form) –

uid = 'select'

```

38.4.1.1.1516 fobi.contrib.plugins.form_elements.fields.select.forms module

```

class fobi.contrib.plugins.form_elements.fields.select.forms.SelectInputForm(data=None,
                                                                              files=None,
                                                                              auto_id='id_%s',
                                                                              pre-
                                                                              fix=None,
                                                                              ini-
                                                                              tial=None,
                                                                              er-
                                                                              ror_class=<class
                                                                              'django.forms.utils.L
                                                                              a-
                                                                              bel_suffix=None,
                                                                              empty_permitted=F
                                                                              field_order=None,
                                                                              use_required_attri

```

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

Form for SelectInputPlugin.

```

base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <django.forms.fields.CharField object>}
clean_initial()
    Validating the initial value.

declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <django.forms.fields.CharField object>}
media
plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''), ('help_text', ''), ('help_text', '')]

```

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

38.4.1.1.1.518 Module contents

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

38.4.1.1.1.520 Submodules

38.4.1.1.1.521 fobi.contrib.plugins.form_elements.fields.select_model_object.apps module

```
class fobi.contrib.plugins.form_elements.fields.select_model_object.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_select_model_object'
    name = 'fobi.contrib.plugins.form_elements.fields.select_model_object'
```

38.4.1.1.1.522 fobi.contrib.plugins.form_elements.fields.select_model_object.base module

```
class fobi.contrib.plugins.form_elements.fields.select_model_object.base.SelectModelObject:
    Bases: fobi.base.FormFieldPlugin
    Select model object field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.select_model_object.
        forms.SelectModelObjectInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                            **kwargs)
        Get form field instances.

    get_queryset()
        Get queryset.

        Might be used in integration packages.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    prepare_plugin_form_data(cleaned_data)
        Prepare plugin form data.

        Might be used in integration plugins.

    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.
```

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

```
uid = 'select_model_object'
```

38.4.1.1.1.523 fobi.contrib.plugins.form_elements.fields.select_model_object.conf module

`fobi.contrib.plugins.form_elements.fields.select_model_object.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_model_object` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

38.4.1.1.1.525 fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements module

class `fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_elements.SelectModelObjectFormFieldPlugin`

Bases: `fobi.base.FormFieldPlugin`

Select model object field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectInputForm`

get_form_field_instances (`request=None`, `form_entry=None`, `form_element_entries=None`, `**kwargs`)

Get form field instances.

get_queryset ()

Get queryset.

Might be used in integration packages.

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

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

prepare_plugin_form_data (`cleaned_data`)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data (`form_entry`, `request`, `form`, `form_element_entries=None`, `**kwargs`)

Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.

```

    • request (django.http.HttpRequest) –
    • form (django.forms.Form) –
uid = 'select_model_object'

```

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

```

class fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObject
    Bases: django.forms.forms.Form, fobi.base.BaseFormFieldPluginForm
    Form for SelectModelObjectPlugin.
    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django
    declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
    media
    plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''), ('

```

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

38.4.1.1.1.528 Module contents

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

38.4.1.1.1.530 Submodules

38.4.1.1.1.531 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps module

```

class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps.Config(app_name, app_model)
    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'

```

38.4.1.1.1.532 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base module

```

class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base.SelectMPTTModelObject
    Bases: fobi.base.FormFieldPlugin
    Select MPTT model object field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None, **kwargs)
        Get form field instances.

```

```
group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

```
uid = 'select_mptt_model_object'
```

38.4.1.1.1.533 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf module

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

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

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

```
class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.fobi_form_elements.
Bases: fobi.base.FormFieldPlugin
```

Select MPTT model object field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectInputForm*

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
**kwargs)
```

Get form field instances.

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

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

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)
 Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'select_mptt_model_object'

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

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

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

Form for *SelectMPTTModelObjectPlugin*.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

media

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

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

38.4.1.1.1.538 Module contents

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

38.4.1.1.1.540 Submodules

38.4.1.1.1.541 *fobi.contrib.plugins.form_elements.fields.select_multiple.apps* module

class *fobi.contrib.plugins.form_elements.fields.select_multiple.apps.Config* (*app_name, app_module*)

Bases: *django.apps.config.AppConfig*

Config.

label = 'fobi_contrib_plugins_form_elements_fields_select_multiple'

name = 'fobi.contrib.plugins.form_elements.fields.select_multiple'

38.4.1.1.1.542 *fobi.contrib.plugins.form_elements.fields.select_multiple.base* module

class *fobi.contrib.plugins.form_elements.fields.select_multiple.base.SelectMultipleInputPluginForm*

Bases: *fobi.base.BaseFormFieldPluginForm*

Select multiple field plugin.

form
 alias of `fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleInputForm`

get_choices()
 Get choices.
 Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)
 Get form field instances.

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

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

prepare_plugin_form_data(*cleaned_data*)
 Prepare plugin form data.
 Might be used in integration plugins.

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)
 Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = `'select_multiple'`

38.4.1.1.1.543 fobi.contrib.plugins.form_elements.fields.select_multiple.conf module

`fobi.contrib.plugins.form_elements.fields.select_multiple.conf.get_setting`(*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_multiple` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

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

class fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.**SelectMultipleFormElement**

Bases: *fobi.base.FormFieldPlugin*

Select multiple field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleInputForm*

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

prepare_plugin_form_data(*cleaned_data*)

Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'select_multiple'

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

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

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

Form for SelectMultipleInputPlugin.

base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <django.forms.fields.CharField object>}

clean_initial()

Validating the initial value.

declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <django.forms.fields.CharField object>}

media

plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''), ('media', '')]

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

38.4.1.1.1.548 Module contents

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

38.4.1.1.1.550 Submodules

38.4.1.1.1.551 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps module

class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps.**Config**(*django.apps.config.AppConfig*)

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'

38.4.1.1.1.552 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.base module

class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.base.**SelectMultipleModelObjectsPlugin**

Bases: *fobi.base.FormFieldPlugin*

Select multiple model objects field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms.SelectMultipleModelObjectsInputForm*

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

get_queryset ()

Get queryset.

Might be used in integration packages.

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

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

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'select_multiple_model_objects'

38.4.1.1.1.553 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf module

fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf.**get_setting** (*setting, override=None*)

Get setting.

Get a setting from *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

38.4.1.1.1.555 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_el

Bases: *fobi.base.FormFieldPlugin*

Select multiple model objects field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.forms.SelectMultipleModelObjectsInputForm*

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

get_queryset()

Get queryset.

Might be used in integration packages.

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

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

submit_plugin_form_data(*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

uid = 'select_multiple_model_objects'

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

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

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

Form for *SelectMultipleModelObjectsPlugin*.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <

media

plugin_data_fields = [('label', ''), ('name', ''), ('model', ''), ('help_text', ''), (

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

38.4.1.1.1.558 Module contents

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

38.4.1.1.1.560 Submodules

38.4.1.1.1.561 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps` module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps.Conf
    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'
```

38.4.1.1.1.562 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.base` module

```
class fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.base.Se
    Bases: fobi.base.FormFieldPlugin
    Select multiple MPTT model object field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms.SelectMultipleMPTTModelObjectsInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
        Submit plugin form data/process.

    Parameters
        • form_entry (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
        • request (django.http.HttpRequest) –
        • form (django.forms.Form) –

    uid = 'select_multiple_mptt_model_objects'
```

38.4.1.1.1.563 `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf` module

`fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf.get_setting`

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.

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

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

class `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_elements`
Bases: `fobi.base.FormFieldPlugin`

Select multiple MPTT model object field plugin.

form

alias of `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms.SelectMultipleMPTTModelObjectsInputForm`

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

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

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

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

uid = 'select_multiple_mptt_model_objects'

38.4.1.1.1.566 fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms module

class fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms.SelectMultipleMPTTModelObjectsForm

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

Form for SelectMultipleMPTTModelObjectsPlugin.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <django.forms.fields.CharField object>}

media

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

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

38.4.1.1.1.568 Module contents

38.4.1.1.1.569 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max package

38.4.1.1.1.570 Submodules

38.4.1.1.1.571 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.apps module

class fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.apps.Config(*app_name*, *app_model*)

Bases: django.apps.config.AppConfig

Config.

label = 'fobi_contrib_plugins_form_elements_fields_select_multiple_with_max'

name = 'fobi.contrib.plugins.form_elements.fields.select_multiple_with_max'

38.4.1.1.1.572 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base module

class fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base.SelectMultipleWithMaxForm

Bases: *fobi.base.FormFieldPlugin*

Select multiple with max field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms.SelectMultipleWithMaxInputForm*

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None*, *form_entry=None*, *form_element_entries=None*, ***kwargs*)

Get form field instances.


```
group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
prepare_plugin_form_data(cleaned_data)
    Prepare plugin form data.

    Might be used in integration plugins.
submit_plugin_form_data(form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

```
uid = 'select_multiple_with_max'
```

38.4.1.1.1.573 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.conf module

```
fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.conf.get_setting(setting,
                                                                                   over-
                                                                                   ride=None)
```

Get setting.

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max` conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

38.4.1.1.1.574 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.defaults module

38.4.1.1.1.575 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fields module

class fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fields.**MultipleChoiceField**

Bases: django.forms.fields.MultipleChoiceField

Multiple choice with max field.

validate (*value*)
Validate.

38.4.1.1.1.576 fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements.**SelectMultipleWithMaxFormPlugin**

Bases: *fobi.base.FormFieldPlugin*

Select multiple with max field plugin.

form
alias of *fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms.SelectMultipleWithMaxInputForm*

get_choices ()
Get choices.

Might be used in integration plugins.

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)
Get form field instances.

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

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

prepare_plugin_form_data (*cleaned_data*)
Prepare plugin form data.

Might be used in integration plugins.

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)
Submit plugin form data/process.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

`uid = 'select_multiple_with_max'`

38.4.1.1.1.577 `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms` module

class `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.forms.SelectMultipl`

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

Form for `SelectMultipleWithMaxInputPlugin`.

base_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <djan

clean_initial()

Validating the initial value.

declared_fields = {'choices': <django.forms.fields.CharField object>, 'help_text': <

media

plugin_data_fields = [('label', ''), ('name', ''), ('choices', ''), ('help_text', ''),

38.4.1.1.1.578 `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.settings` module

38.4.1.1.1.579 Module contents

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

38.4.1.1.1.581 Submodules

38.4.1.1.1.582 `fobi.contrib.plugins.form_elements.fields.slider.apps` module

class `fobi.contrib.plugins.form_elements.fields.slider.apps.Config`(*app_name*,
app_module)

Bases: `django.apps.config.AppConfig`

`Config`.

```
label = 'fobi_contrib_plugins_form_elements_fields_slider'
name = 'fobi.contrib.plugins.form_elements.fields.slider'
```

38.4.1.1.1.583 fobi.contrib.plugins.form_elements.fields.slider.base module

```
class fobi.contrib.plugins.form_elements.fields.slider.base.SliderInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin
    Slider field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.slider.forms.SliderInputForm

    get_choices()
        Get choices.

        Might be used in integration plugins.

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                               **kwargs)
        Get form field instances.

    get_initial()
        Get initial value.

        Might be used in integration plugins.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    html_classes = ['slider']
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'slider'
```

38.4.1.1.1.584 fobi.contrib.plugins.form_elements.fields.slider.conf module

```
fobi.contrib.plugins.form_elements.fields.slider.conf.get_setting (setting,
                                                                    over-
                                                                    ride=None)

    Get setting.

    Get a setting from fobi.contrib.plugins.form_elements.fields.slider conf module, falling back to the default.

    If override is not None, it will be used instead of the setting.
```

Parameters

- **setting** – String with setting name.
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

38.4.1.1.1.585 fobi.contrib.plugins.form_elements.fields.slider.constants module

38.4.1.1.1.586 fobi.contrib.plugins.form_elements.fields.slider.defaults module

38.4.1.1.1.587 fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements module

class fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements.SliderInputPlugin

Bases: *fobi.base.FormFieldPlugin*

Slider field plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.slider.forms.SliderInputForm*

get_choices()

Get choices.

Might be used in integration plugins.

get_form_field_instances(*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get form field instances.

get_initial()

Get initial value.

Might be used in integration plugins.

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

html_classes = ['slider']

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

uid = 'slider'

38.4.1.1.1.588 fobi.contrib.plugins.form_elements.fields.slider.forms module

class fobi.contrib.plugins.form_elements.fields.slider.forms.SliderInputForm(*data=None,*

files=None,

auto_id='id_%s',

pre-

fix=None,

ini-

tial=None,

er-

ror_class=<class

'django.forms.utils.l

la-

bel_suffix=None,

empty_permitted=F

field_order=None,

use_required_attri

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

Form for SliderInputPlugin.

base_fields = {'custom_ticks': <django.forms.fields.CharField object>, 'handle': <dj

```

clean()
    Validating the values.

declared_fields = {'custom_ticks': <django.forms.fields.CharField object>, 'handle':
media
plugin_data_fields = [('label', ''), ('name', ''), ('initial', 50), ('min_value', 0),

```

38.4.1.1.1.589 fobi.contrib.plugins.form_elements.fields.slider.helpers module

fobi.contrib.plugins.form_elements.fields.slider.helpers.**generate_ticks** (*choices*,
empty_labels=False)

Generate ticks.

Parameters

- **choices** (*iterable*) – Iterable of tuples or lists:
- **empty_labels** (*bool*) –

Return dict

38.4.1.1.1.590 fobi.contrib.plugins.form_elements.fields.slider.settings module

38.4.1.1.1.591 fobi.contrib.plugins.form_elements.fields.slider.widgets module

```

class fobi.contrib.plugins.form_elements.fields.slider.widgets.BaseSliderPluginWidget (plugin
Bases: fobi.base.FormElementPluginWidget
Base date form element plugin widget.

html_classes = ['slider']

plugin_uid = 'slider'

```

38.4.1.1.1.592 Module contents

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

38.4.1.1.1.594 Submodules

38.4.1.1.1.595 fobi.contrib.plugins.form_elements.fields.slug.apps module

```

class fobi.contrib.plugins.form_elements.fields.slug.apps.Config (app_name,
app_module)
Bases: django.apps.config.AppConfig
Config.

label = 'fobi_contrib_plugins_form_elements_fields_slug'

name = 'fobi.contrib.plugins.form_elements.fields.slug'

```

38.4.1.1.1.596 fobi.contrib.plugins.form_elements.fields.slug.base module

```
class fobi.contrib.plugins.form_elements.fields.slug.base.SlugInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Slug field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.slug.forms.
                             SlugInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'slug'
```

38.4.1.1.1.597 fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin

    Slug field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.slug.forms.
                             SlugInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'slug'
```

38.4.1.1.1.598 fobi.contrib.plugins.form_elements.fields.slug.forms module

```
class fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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 SlugInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan
```

```
clean()
```

Hook for doing any extra form-wide cleaning after `Field.clean()` has been called on every field. Any `ValidationError` raised by this method will not be associated with a particular field; it will have a special-case association with the field named `'__all__'`.

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
```

```
media
```

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

38.4.1.1.1.599 Module contents

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

38.4.1.1.1.601 Submodules

38.4.1.1.1.602 fobi.contrib.plugins.form_elements.fields.text.apps module

```
class fobi.contrib.plugins.form_elements.fields.text.apps.Config(app_name,
                                                                    app_module)
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_fields_text'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.text'
```

38.4.1.1.1.603 fobi.contrib.plugins.form_elements.fields.text.base module

```
class fobi.contrib.plugins.form_elements.fields.text.base.TextInputPlugin(user=None)
```

Bases: `fobi.base.BaseFormFieldPlugin`

Text field plugin.

```

form
    alias      of      fobi.contrib.plugins.form_elements.fields.text.forms.
                        TextInputForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'text'

```

38.4.1.1.1.604 fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements module

```

class fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin (use
    Bases: fobi.base.FormFieldPlugin

```

Text field plugin.

```

form
    alias      of      fobi.contrib.plugins.form_elements.fields.text.forms.
                        TextInputForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'text'

```

38.4.1.1.1.605 fobi.contrib.plugins.form_elements.fields.text.forms module

```

class fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm (data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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 TextInputPlugin.

```

base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan

```

```

clean()
    Validation.

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''),

```

38.4.1.1.1.606 Module contents

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

38.4.1.1.1.608 Submodules

38.4.1.1.1.609 fobi.contrib.plugins.form_elements.fields.textarea.apps module

```

class fobi.contrib.plugins.form_elements.fields.textarea.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_fields_textarea'
    name = 'fobi.contrib.plugins.form_elements.fields.textarea'

```

38.4.1.1.1.610 fobi.contrib.plugins.form_elements.fields.textarea.base module

```

class fobi.contrib.plugins.form_elements.fields.textarea.base.TextareaPlugin(user=None)
    Bases: fobi.base.FormFieldPlugin
    Textarea field plugin.

    form
        alias      of      fobi.contrib.plugins.form_elements.fields.textarea.forms.
                           TextareaForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'textarea'

```

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

```

class fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements.TextareaPlugin
    Bases: fobi.base.FormFieldPlugin
    Textarea field plugin.

```

```

form
    alias      of      fobi.contrib.plugins.form_elements.fields.textarea.forms.
                        TextareaForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'textarea'

```

38.4.1.1.1.612 fobi.contrib.plugins.form_elements.fields.textarea.forms module

```

class fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm (data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           pre-
                                                                           fix=None,
                                                                           ini-
                                                                           tial=None,
                                                                           er-
                                                                           ror_class=<class
                                                                           'django.forms.utils.E
                                                                           la-
                                                                           bel_suffix=None,
                                                                           empty_permitted=Fa
                                                                           field_order=None,
                                                                           use_required_attribu

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

    base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan
    declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
    media
    plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''),

```

38.4.1.1.1.613 Module contents

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

38.4.1.1.1.615 Submodules

38.4.1.1.1.616 fobi.contrib.plugins.form_elements.fields.time.apps module

```

class fobi.contrib.plugins.form_elements.fields.time.apps.Config (app_name,
                                                                    app_module)

    Bases: django.apps.config.AppConfig
    Config.

```

```
label = 'fobi_contrib_plugins_form_elements_fields_time'
name = 'fobi.contrib.plugins.form_elements.fields.time'
```

38.4.1.1.1.617 fobi.contrib.plugins.form_elements.fields.time.base module

```
class fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin
```

Time field plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.time.forms.
                       TimeInputForm
```

```
get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
```

Get form field instances.

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

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

```
submit_plugin_form_data (form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

```
uid = 'time'
```

38.4.1.1.1.618 fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin (user=None)
    Bases: fobi.base.FormFieldPlugin
```

Time field plugin.

```
form
    alias      of      fobi.contrib.plugins.form_elements.fields.time.forms.
                       TimeInputForm
```

```
get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                           **kwargs)
```

Get form field instances.

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

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

```
submit_plugin_form_data (form_entry, request, form, form_element_entries=None, **kwargs)
    Submit plugin form data/process.
```

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

```
uid = 'time'
```

38.4.1.1.1.619 `fobi.contrib.plugins.form_elements.fields.time.forms` module

```
class fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           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 TimeInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan
clean_initial()
    Clean the initial value.

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''),
```

38.4.1.1.1.620 Module contents

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

38.4.1.1.1.622 Submodules

38.4.1.1.1.623 `fobi.contrib.plugins.form_elements.fields.url.apps` module

```
class fobi.contrib.plugins.form_elements.fields.url.apps.Config(app_name,
                                                                app_module)

Bases: django.apps.config.AppConfig
Config.

label = 'fobi_contrib_plugins_form_elements_fields_url'
```

```
name = 'fobi.contrib.plugins.form_elements.fields.url'
```

38.4.1.1.1.624 fobi.contrib.plugins.form_elements.fields.url.base module

```
class fobi.contrib.plugins.form_elements.fields.url.base.URLInputPlugin(user=None)
```

Bases: *fobi.base.FormFieldPlugin*

URL input plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm*

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None,  
                           **kwargs)
```

Get form field instances.

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

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

```
uid = 'url'
```

38.4.1.1.1.625 fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin(user=None)
```

Bases: *fobi.base.FormFieldPlugin*

URL input plugin.

form

alias of *fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm*

```
get_form_field_instances(request=None, form_entry=None, form_element_entries=None,  
                           **kwargs)
```

Get form field instances.

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

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

```
uid = 'url'
```

38.4.1.1.1.626 fobi.contrib.plugins.form_elements.fields.url.forms module

```
class fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm(data=None,
                                                                    files=None,
                                                                    auto_id='id_%s',
                                                                    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 = {'help_text': <django.forms.fields.CharField object>, 'initial': <djan
clean()
    Hook for doing any extra form-wide cleaning after Field.clean() has been called on every field. Any
    ValidationError raised by this method will not be associated with a particular field; it will have a special-
    case association with the field named '__all__'.

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'initial': <
media
plugin_data_fields = [('label', ''), ('name', ''), ('help_text', ''), ('initial', ''),
```

38.4.1.1.1.627 Module contents

38.4.1.1.1.628 Module contents

38.4.1.1.1.629 fobi.contrib.plugins.form_elements.security package

38.4.1.1.1.630 Subpackages

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

38.4.1.1.1.632 Submodules

38.4.1.1.1.633 fobi.contrib.plugins.form_elements.security.captcha.apps module

```
class fobi.contrib.plugins.form_elements.security.captcha.apps.Config(app_name,
                                                                    app_module)

Bases: django.apps.config.AppConfig
Config.
label = 'fobi_contrib_plugins_form_elements_security_captcha'
```

```
name = 'fobi.contrib.plugins.form_elements.security.captcha'
```

38.4.1.1.1.634 fobi.contrib.plugins.form_elements.security.captcha.base module

```
class fobi.contrib.plugins.form_elements.security.captcha.base.CaptchaInputPlugin (user=None)
```

Bases: *fobi.base.FormElementPlugin*

Captcha field plugin.

form

alias of *fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaInputForm*

```
get_form_field_instances (request=None, form_entry=None, form_element_entries=None,  
                           **kwargs)
```

Get form field instances.

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

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

```
uid = 'captcha'
```

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

```
class fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaInputP
```

Bases: *fobi.base.FormElementPlugin*

Captcha field plugin.

form

alias of *fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaInputForm*

```
get_form_field_instances (request=None, form_entry=None, form_element_entries=None,  
                           **kwargs)
```

Get form field instances.

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

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

```
uid = 'captcha'
```


38.4.1.1.1.636 fobi.contrib.plugins.form_elements.security.captcha.forms module

```
class fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaInputForm(data=None,
                                     files=None,
                                     auto_id='id_%d',
                                     pre-
                                     fix=None,
                                     ini-
                                     tial=None,
                                     er-
                                     ror_class=<django.forms
                                     'django.forms
                                     la-
                                     bel_suffix=None,
                                     empty_permit-
                                     field_order=None,
                                     use_required_
```

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

Form for CaptchaInputPlugin.

```
base_fields = {'help_text': <django.forms.fields.CharField object>, 'label': <django
```

```
declared_fields = {'help_text': <django.forms.fields.CharField object>, 'label': <dj
```

```
media
```

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

38.4.1.1.1.637 Module contents

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

38.4.1.1.1.639 Submodules

38.4.1.1.1.640 fobi.contrib.plugins.form_elements.security.honeypot.apps module

```
class fobi.contrib.plugins.form_elements.security.honeypot.apps.Config(app_name,
                                                                    app_module)
```

Bases: `django.apps.config AppConfig`

Config.

```
label = 'fobi_contrib_plugins_form_elements_security_honeypot'
```

```
name = 'fobi.contrib.plugins.form_elements.security.honeypot'
```

38.4.1.1.1.641 fobi.contrib.plugins.form_elements.security.honeypot.base module

```
class fobi.contrib.plugins.form_elements.security.honeypot.base.HoneypotInputPlugin(user=None)
```

Bases: `fobi.base.FormElementPlugin`

Honeypot field plugin.

```

form
    alias of fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotInputForm

get_form_field_instances (request=None, form_entry=None, form_element_entries=None, **kwargs)
    Get form field instances.

group = <django.utils.functional.lazy.<locals>.__proxy__ object>
is_hidden = True
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'honeypot'

```

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

```

fobi.contrib.plugins.form_elements.security.honeypot.conf.get_setting (setting, override=None)

    Get setting.

    Get a setting from fobi.contrib.plugins.form_elements.security.honeypot conf module,
    falling back to the default.

    If override is not None, it will be used instead of the setting.

```

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

38.4.1.1.1.644 fobi.contrib.plugins.form_elements.security.honeypot.fields module

```

class fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotField (max_length=None, min_length=None, strip=True, *args, **kwargs)

    Bases: django.forms.fields.CharField

clean (value)
    Check that honeypot value remained the same.

default_error_messages = {'invalid': <django.utils.functional.lazy.<locals>.__proxy__ object>}

widget
    alias of django.forms.widgets.HiddenInput

```

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

```
class fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin
    Bases: fobi.base.FormElementPlugin

    Honeypot field plugin.

    form
        alias of fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotInputForm

    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.

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

    is_hidden = True

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

    uid = 'honeypot'
```

38.4.1.1.1.646 fobi.contrib.plugins.form_elements.security.honeypot.forms module

```
class fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotInputForm (data=None, files=None,
auto_id='id', initial=None, label=None, max_length=None, min_length=None, pre-
fix=None, postfix=None, required=True, error_class=django.forms.errors.ValidationError, label_suffix=' ',
empty_permitted=False, field_order=None, use_required_attribute=True)

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

    Form for HoneypotInputPlugin.

    base_fields = {'initial': <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField object>}

    declared_fields = {'initial': <django.forms.fields.CharField object>, 'label': <django.forms.fields.CharField object>}

    media
        django.forms.media.Media

    plugin_data_fields = [('label', ''), ('name', ''), ('initial', ''), ('max_length', '255'), ('min_length', '3')]

    required = True
```

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

- HONEYPOT_VALUE (string)

38.4.1.1.1.648 Module contents

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

38.4.1.1.1.650 Submodules

38.4.1.1.1.651 fobi.contrib.plugins.form_elements.security.recaptcha.apps module

```
class fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_elements_security_recaptcha'
    name = 'fobi.contrib.plugins.form_elements.security.recaptcha'
```

38.4.1.1.1.652 fobi.contrib.plugins.form_elements.security.recaptcha.base module

```
class fobi.contrib.plugins.form_elements.security.recaptcha.base.ReCaptchaInputPlugin(user=I
    Bases: fobi.base.FormElementPlugin
    ReCaptcha field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.security.recaptcha.forms.
        ReCaptchaInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'recaptcha'
```

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

```
class fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements.ReCaptchaInp
    Bases: fobi.base.FormElementPlugin
    ReCaptcha field plugin.
    form
        alias of fobi.contrib.plugins.form_elements.security.recaptcha.forms.
        ReCaptchaInputForm
    get_form_field_instances(request=None, form_entry=None, form_element_entries=None,
                             **kwargs)
        Get form field instances.
    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    uid = 'recaptcha'
```

38.4.1.1.1.654 fobi.contrib.plugins.form_elements.security.recaptcha.forms module

class fobi.contrib.plugins.form_elements.security.recaptcha.forms.**ReCaptchaInputForm** (*data=No*

files=No
auto_id=
pre-
fix=Non
ini-
tial=No
er-
ror_clas
'django.
la-
bel_suff
empty_p
field_or
use_req

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

Form for ReCaptchaInputPlugin.

base_fields = {'help_text': <django.forms.fields.CharField object>, 'label': <django

declared_fields = {'help_text': <django.forms.fields.CharField object>, 'label': <dj

media

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

38.4.1.1.1.655 Module contents

38.4.1.1.1.656 Module contents

38.4.1.1.1.657 fobi.contrib.plugins.form_elements.test package

38.4.1.1.1.658 Subpackages

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

38.4.1.1.1.660 Submodules

38.4.1.1.1.661 fobi.contrib.plugins.form_elements.test.dummy.apps module

class fobi.contrib.plugins.form_elements.test.dummy.apps.**Config** (*app_name,*
app_module)

Bases: django.apps.config.AppConfig

Config.

label = 'fobi_contrib_plugins_form_elements_test_dummy'

name = 'fobi.contrib.plugins.form_elements.test.dummy'

38.4.1.1.1.662 fobi.contrib.plugins.form_elements.test.dummy.base module

```
class fobi.contrib.plugins.form_elements.test.dummy.base.DummyPlugin (user=None)
    Bases: fobi.base.FormElementPlugin

    Dummy plugin.

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                               **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    post_processor ()
        Post process data.

        Always the same.

    uid = 'dummy'
```

38.4.1.1.1.663 fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements module

```
class fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin (user=None)
    Bases: fobi.base.FormElementPlugin

    Dummy plugin.

    get_form_field_instances (request=None, form_entry=None, form_element_entries=None,
                               **kwargs)
        Get form field instances.

    group = <django.utils.functional.lazy.<locals>.__proxy__ object>
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>

    post_processor ()
        Post process data.

        Always the same.

    uid = 'dummy'
```

38.4.1.1.1.664 fobi.contrib.plugins.form_elements.test.dummy.widgets module

```
class fobi.contrib.plugins.form_elements.test.dummy.widgets.BaseDummyPluginWidget (plugin)
    Bases: fobi.base.FormElementPluginWidget

    Base dummy form element plugin widget.

    plugin_uid = 'dummy'
```

38.4.1.1.1.665 Module contents

38.4.1.1.1.666 Module contents

38.4.1.1.1.667 Module contents

38.4.1.1.1.668 fobi.contrib.plugins.form_handlers package

38.4.1.1.1.669 Subpackages

38.4.1.1.1.670 fobi.contrib.plugins.form_handlers.db_store package

38.4.1.1.1.671 Subpackages

38.4.1.1.1.672 fobi.contrib.plugins.form_handlers.db_store.migrations package

38.4.1.1.1.673 Submodules

38.4.1.1.1.674 fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial module

```
class fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial.Migration(name,
                                     app_label,
                                     Bases: django.db.migrations.migration.Migration
                                     dependencies = [('fobi', '0002_auto_20150912_1744'), ('auth', '__first__')]
                                     operations = [<CreateModel name='SavedFormDataEntry', fields=[('id', <django.db.models
```

38.4.1.1.1.675 fobi.contrib.plugins.form_handlers.db_store.migrations.0002_savedformwizarddataentry module

```
class fobi.contrib.plugins.form_handlers.db_store.migrations.0002_savedformwizarddataentry
                                     Bases: django.db.migrations.migration.Migration
                                     dependencies = [('auth', '__first__'), ('fobi', '0010_formwizardhandler'), ('fobi_cont
                                     operations = [<CreateModel name='SavedFormWizardDataEntry', fields=[('id', <django.db.
```

38.4.1.1.1.676 Module contents

38.4.1.1.1.677 fobi.contrib.plugins.form_handlers.db_store.urls package

38.4.1.1.1.678 Submodules

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

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

38.4.1.1.1.681 Module contents

38.4.1.1.1.682 Submodules

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

```
class fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDataEntryAdmin(model,
                                                                              ad-
                                                                              min_site)
    Bases: fobi.contrib.plugins.form_handlers.db_store.admin.
            BaseSavedFormDataEntryAdmin
    Saved form data entry admin.

    class Meta
        Bases: object
        Meta class.

        app_label = <django.utils.functional.lazy.<locals>.__proxy__ object>
    actions = ['export_data']
    fieldsets = ((None, {'fields': ('form_entry', 'user')}), (<django.utils.functional.la
    list_display = ('form_entry', 'user', 'formatted_saved_data', 'created')
    list_filter = ('form_entry', 'user')
    media
    only_args = ['form_entry']
    readonly_fields = ('created', 'formatted_saved_data')

class fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntryAdmin(model,
                                                                              ad-
                                                                              min_si

    Bases: fobi.contrib.plugins.form_handlers.db_store.admin.
            BaseSavedFormDataEntryAdmin
    Saved form wizard data entry admin.

    class Meta
        Bases: object
        Meta class.

        app_label = <django.utils.functional.lazy.<locals>.__proxy__ object>
```



```

actions = ['export_data']
fieldsets = (None, {'fields': ('form_wizard_entry', 'user')}), (<django.utils.function
list_display = ('form_wizard_entry', 'user', 'formatted_saved_data', 'created')
list_filter = ('form_wizard_entry', 'user')
media
only_args = ['form_wizard_entry']
readonly_fields = ('created', 'formatted_saved_data')

```

38.4.1.1.1.684 fobi.contrib.plugins.form_handlers.db_store.apps module

```

class fobi.contrib.plugins.form_handlers.db_store.apps.Config(app_name,
                                                             app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_handlers_db_store'
    name = 'fobi.contrib.plugins.form_handlers.db_store'

```

38.4.1.1.1.685 fobi.contrib.plugins.form_handlers.db_store.base module

```

class fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin(user=None)
    Bases: fobi.base.FormHandlerPlugin
    DB store form handler plugin.
    Can be used only once per form.
    allow_multiple = False
    custom_actions (form_entry, request=None)
        Custom actions.
        Adding a link to view the saved form entries.
        Return iterable
    name = <django.utils.functional.lazy.<locals>.__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.

save_form_data_entry (*form_entry, request, field_name_to_label_map, cleaned_data*)

Save form data entry.

Might be used in integration plugins.

uid = 'db_store'

class `fobi.contrib.plugins.form_handlers.db_store.base.DBStoreWizardHandlerPlugin` (*user=None*)

Bases: `fobi.base.FormWizardHandlerPlugin`

DB store form wizard handler plugin.

Can be used only once per form.

allow_multiple = **False**

custom_actions (*form_wizard_entry, request=None*)

Custom actions.

Adding a link to view the saved form entries.

Return iterable

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

run (*form_wizard_entry, request, form_list, form_wizard, form_element_entries=None*)

Run.

Parameters

- **form_wizard_entry** (`fobi.models.FormWizardEntry`) – Instance of `fobi.models.FormWizardEntry`.
- **request** (`django.http.HttpRequest`) –
- **form_list** (*list*) – List of `django.forms.Form` instances.
- **form_wizard** (`fobi.wizard.views.dynamic.DynamicWizardView`) – Instance of `fobi.wizard.views.dynamic.DynamicWizardView`.
- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry` objects.

uid = 'db_store'

38.4.1.1.1.686 fobi.contrib.plugins.form_handlers.db_store.callbacks module

class `fobi.contrib.plugins.form_handlers.db_store.callbacks.AutoFormDbStore`

Bases: `fobi.base.FormCallback`

Auto save form entries.

Note, that this callback is not active. In order to activate it, you should import the `AutoFormDbStore` and register it using the callback register as follows.

```
>>> from fobi.base import form_callback_registry
>>> from fobi.contrib.plugins.form_handlers.db_store.callbacks import (
>>>     AutoFormDbStore
>>> )
>>> form_callback_registry.register(AutoFormDbStore)
```

callback (*form_entry, request, form*)

Callback.

Parameters

- **form_entry** –
- **request** –
- **form** –

Returns

stage = 'form_valid'

38.4.1.1.1.687 fobi.contrib.plugins.form_handlers.db_store.conf module

`fobi.contrib.plugins.form_handlers.db_store.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from `fobi.contrib.plugins.form_handlers.db_store.conf` module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

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

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

class `fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreHandlerPlugin` (`fobi.base.FormHandlerPlugin`)

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.lazy.<locals>.__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.

save_form_data_entry (*form_entry, request, field_name_to_label_map, cleaned_data*)
 Save form data entry.

Might be used in integration plugins.

uid = 'db_store'

class *fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.DBStoreWizardHandlerPlugin*
 Bases: *fobi.base.FormWizardHandlerPlugin*

DB store form wizard handler plugin.

Can be used only once per form.

allow_multiple = **False**

custom_actions (*form_wizard_entry, request=None*)
 Custom actions.

Adding a link to view the saved form entries.

Return iterable

name = <*django.utils.functional.lazy.<locals>.__proxy__ object*>

run (*form_wizard_entry, request, form_list, form_wizard, form_element_entries=None*)
 Run.

Parameters

- **form_wizard_entry** (*fobi.models.FormWizardEntry*) – Instance of *fobi.models.FormWizardEntry*.
- **request** (*django.http.HttpRequest*) –
- **form_list** (*list*) – List of *django.forms.Form* instances.
- **form_wizard** (*fobi.wizard.views.dynamic.DynamicWizardView*) – Instance of *fobi.wizard.views.dynamic.DynamicWizardView*.
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

uid = 'db_store'

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

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

Bases: *object*

Exporting the data.

export_to_csv ()
 Export data to CSV.

export_to_xls ()
 Export data to XLS.

graceful_export()

Export data into XLS/CSV depending on what is available.

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

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

Bases: django.db.models.base.Model

Abstract saved form data entry.

class Meta

Bases: object

Meta options.

abstract = False

created

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

form_data_headers

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

formatted_saved_data()

Shows the formatted saved data records.

Return string

get_next_by_created (**morekwargs)

get_previous_by_created (**morekwargs)

saved_data

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

user

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

In the example:

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

child.parent is a ForwardManyToOneDescriptor instance.

user_id

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

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

Bases: *fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataRowEntry*

Saved form data.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_entry

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

In the example:

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

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

form_entry_id

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

get_next_by_created (***morekwargs*)

get_previous_by_created (***morekwargs*)

id

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

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

user

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

In the example:

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

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

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

Bases: `fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataEntry`

Saved form data.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_wizard_entry

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

In the example:

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

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

form_wizard_entry_id

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

`get_next_by_created(**morekwargs)`

`get_previous_by_created(**morekwargs)`

`id`

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

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

`user`

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

In the example:

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

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

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

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

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

`fobi.contrib.plugins.form_handlers.db_store.views.view_saved_form_data_entries` (*request*, *form_entry_id=None*, *theme=None*, *template_name='db_store/entries.html'*)

View saved form data entries.

Parameters

- `request` (*django.http.HttpRequest*) –
- `form_entry_id` (*int*) – Form ID.
- `theme` (*fobi.base.BaseTheme*) – Subclass of `fobi.base.BaseTheme`.
- `template_name` (*string*) –

Return `django.http.HttpResponse`

`fobi.contrib.plugins.form_handlers.db_store.views.export_saved_form_data_entries` (*request*, *form_entry_id=None*, *theme=None*)

Export saved form data entries.

Parameters

- `request` (*django.http.HttpRequest*) –
- `form_entry_id` (*int*) – Form ID.
- `theme` (*fobi.base.BaseTheme*) – Subclass of `fobi.base.BaseTheme`.

Return `django.http.HttpResponse`

`fobi.contrib.plugins.form_handlers.db_store.views.view_saved_form_wizard_data_entries` (*request*, *form_id*, *theme*, *template_name*)

View saved form wizard data entries.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_wizard_entry_id** (*int*) – Form ID.
- **theme** (*fobi.base.BaseTheme*) – Subclass of `fobi.base.BaseTheme`.
- **template_name** (*string*) –

Return `django.http.HttpResponse`

`fobi.contrib.plugins.form_handlers.db_store.views.export_saved_form_wizard_data_entries` (*request*, *form_id*, *theme*)

Export saved form wizard data entries.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_wizard_entry_id** (*int*) – Form ID.
- **theme** (*fobi.base.BaseTheme*) – Subclass of `fobi.base.BaseTheme`.

Return `django.http.HttpResponse`

38.4.1.1.1.694 fobi.contrib.plugins.form_handlers.db_store.widgets module

```
class fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget (plugin)
    Bases: fobi.base.FormHandlerPluginWidget
    Base dummy form element plugin widget.

    export_entries_icon_class = 'glyphicon glyphicon-export'
    plugin_uid = 'db_store'
    view_entries_icon_class = 'glyphicon glyphicon-list'
    view_saved_form_data_entries_template_name = 'db_store/view_saved_form_data_entries.html'
```

38.4.1.1.1.695 Module contents

38.4.1.1.1.696 fobi.contrib.plugins.form_handlers.http_repost package

38.4.1.1.1.697 Submodules

38.4.1.1.1.698 fobi.contrib.plugins.form_handlers.http_repost.apps module

```
class fobi.contrib.plugins.form_handlers.http_repost.apps.Config (app_name, app_module)
    Bases: django.apps.config.AppConfig
```


Config.

```
label = 'fobi_contrib_plugins_form_handlers_http_repost'
name = 'fobi.contrib.plugins.form_handlers.http_repost'
```

38.4.1.1.1.699 fobi.contrib.plugins.form_handlers.http_repost.base module

class `fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandlerPlugin` (*user=None*)
 Bases: `fobi.base.FormHandlerPlugin`

HTTP repost handler plugin.

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

do_http_repost (*request, files*)

Re-post data via HTTP.

Might be used in integration plugins.

form

alias of `fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm`

name = `<django.utils.functional.lazy.<locals>.__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'`

class `fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostWizardHandlerPlugin` (*use*)
 Bases: `fobi.base.FormWizardHandlerPlugin`

HTTP repost wizard handler plugin.

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

form

alias of `fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm`

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

plugin_data_repr ()

Human readable representation of plugin data.

Return string

run (*form_wizard_entry*, *request*, *form_list*, *form_wizard*, *form_element_entries=None*)
Run.

Parameters

- **form_wizard_entry** (*fobi.models.FormWizardEntry*) – Instance of *fobi.models.FormWizardEntry*.
- **request** (*django.http.HttpRequest*) –
- **form_list** (*list*) – List of *django.forms.Form* instances.
- **form_wizard** (*fobi.wizard.views.dynamic.DynamicWizardView*) – Instance of *fobi.wizard.views.dynamic.DynamicWizardView*.
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

uid = 'http_repost'

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

class *fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostHandlerPlugin*
Bases: *fobi.base.FormHandlerPlugin*

HTTP repost handler plugin.

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

do_http_repost (*request*, *files*)

Re-post data via HTTP.

Might be used in integration plugins.

form

alias of *fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm*

name = <*django.utils.functional.lazy.<locals>.__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'

```
class fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostWizardHandler
    Bases: fobi.base.FormWizardHandlerPlugin

    HTTP repost wizard handler plugin.

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

    form
        alias      of      fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm

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

    plugin_data_repr()
        Human readable representation of plugin data.

        Return string

    run(form_wizard_entry, request, form_list, form_wizard, form_element_entries=None)
        Run.

        Parameters

        • form_wizard_entry (fobi.models.FormWizardEntry) – Instance of fobi.models.FormWizardEntry.

        • request (django.http.HttpRequest) –

        • form_list (list) – List of django.forms.Form instances.

        • form_wizard (fobi.wizard.views.dynamic.DynamicWizardView) – Instance of fobi.wizard.views.dynamic.DynamicWizardView.

        • form_element_entries (iterable) – Iterable of fobi.models.FormElementEntry objects.

    uid = 'http_repost'
```

38.4.1.1.1.701 fobi.contrib.plugins.form_handlers.http_repost.forms module

```
class fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm(data=None,
                                                                           files=None,
                                                                           auto_id='id_%s',
                                                                           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.BasePluginForm

    Form for HTTPRepostPlugin.

    base_fields = {'endpoint_url': <django.forms.fields.URLField object>}

    declared_fields = {'endpoint_url': <django.forms.fields.URLField object>}
```

```
media
plugin_data_fields = [('endpoint_url', '')]
```

38.4.1.1.1.702 Module contents

38.4.1.1.1.703 fobi.contrib.plugins.form_handlers.mail package

38.4.1.1.1.704 Submodules

38.4.1.1.1.705 fobi.contrib.plugins.form_handlers.mail.apps module

```
class fobi.contrib.plugins.form_handlers.mail.apps.Config(app_name,
                                                         app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_handlers_mail'
    name = 'fobi.contrib.plugins.form_handlers.mail'
```

38.4.1.1.1.706 fobi.contrib.plugins.form_handlers.mail.base module

```
class fobi.contrib.plugins.form_handlers.mail.base.MailHandlerPlugin(user=None)
    Bases: fobi.base.FormHandlerPlugin, fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin
    Mail handler plugin.
    Sends emails to the person specified. Should be executed before db_store and http_repost plugins.
    form
        alias of fobi.contrib.plugins.form_handlers.mail.forms.MailForm
    name = <django.utils.functional.lazy.<locals>.__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'
```

```
class fobi.contrib.plugins.form_handlers.mail.base.MailWizardHandlerPlugin (user=None)
    Bases: fobi.base.FormWizardHandlerPlugin

    Mail wizard handler plugin.

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

    form
        alias of fobi.contrib.plugins.form_handlers.mail.forms.MailForm

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

    plugin_data_repr()
        Human readable representation of plugin data.

        Return string

    run (form_wizard_entry, request, form_list, form_wizard, form_element_entries=None)
        Run.

    Parameters

    • form_wizard_entry (fobi.models.FormWizardEntry) – Instance of fobi.models.FormWizardEntry.

    • request (django.http.HttpRequest) –

    • form_list (list) – List of django.forms.Form instances.

    • form_wizard (fobi.wizard.views.dynamic.DynamicWizardView) – Instance of fobi.wizard.views.dynamic.DynamicWizardView.

    • form_element_entries (iterable) – Iterable of fobi.models.FormElementEntry objects.

    uid = 'mail'
```

38.4.1.1.1.707 fobi.contrib.plugins.form_handlers.mail.callbacks module

```
class fobi.contrib.plugins.form_handlers.mail.callbacks.AutoFormMail
    Bases: fobi.base.FormCallback, fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin
```

Auto mail form entries.

Note, that this callback is not active. In order to activate it, you should import the `AutoFormMail` and register it using the callback register as follows.

```
>>> from fobi.base import form_callback_registry
>>> from fobi.contrib.plugins.form_handlers.mail.callbacks import (
>>>     AutoFormMail
>>> )
>>> form_callback_registry.register(AutoFormMail)
```

```
callback (form_entry, request, form)
    Callback.
```

Parameters

- **form_entry** –
- **request** –
- **form** –

Returns

send_email (*rendered_data, files*)

Send email.

Might be used in integration packages.

stage = 'form_valid'

38.4.1.1.1.708 fobi.contrib.plugins.form_handlers.mail.conf module

fobi.contrib.plugins.form_handlers.mail.conf.get_setting (*setting, override=None*)

Get setting.

Get a setting from `fobi.contrib.plugins.form_handlers.mail.conf` module, falling back to the default.

If `override` is not `None`, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to `None`.

Returns Setting value.

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

38.4.1.1.1.710 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="",
                                                                    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.lazy.<locals>.__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 *fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget*

38.4.1.1.1.711 fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers module

class *fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailHandlerPlugin* (*user=None*)
 Bases: *fobi.base.FormHandlerPlugin*, *fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin*

Mail handler plugin.

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

form
 alias of *fobi.contrib.plugins.form_handlers.mail.forms.MailForm*

name = <django.utils.functional.lazy.<locals>.__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'

class *fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailWizardHandlerPlugin* (*us*)
 Bases: *fobi.base.FormWizardHandlerPlugin*

Mail wizard handler plugin.

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

form
 alias of *fobi.contrib.plugins.form_handlers.mail.forms.MailForm*

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

plugin_data_repr ()
 Human readable representation of plugin data.

Return string

```
run(form_wizard_entry, request, form_list, form_wizard, form_element_entries=None)
Run.
```

Parameters

- **form_wizard_entry** (`fobi.models.FormWizardEntry`) – Instance of `fobi.models.FormWizardEntry`.
- **request** (`django.http.HttpRequest`) –
- **form_list** (`list`) – List of `django.forms.Form` instances.
- **form_wizard** (`fobi.wizard.views.dynamic.DynamicWizardView`) – Instance of `fobi.wizard.views.dynamic.DynamicWizardView`.
- **form_element_entries** (`iterable`) – Iterable of `fobi.models.FormElementEntry` objects.

```
uid = 'mail'
```

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

```
class fobi.contrib.plugins.form_handlers.mail.forms.MailForm(data=None,
                                                            files=None,
                                                            auto_id='id_%s',
                                                            prefix=None,
                                                            initial=None, error_class=<class
'django.forms.utils.ErrorList'>,
                                                            label_suffix=None,
                                                            empty_permitted=False,
                                                            field_order=None,
                                                            use_required_attribute=None)
```

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

Form for BooleanSelectPlugin.

```
base_fields = {'body': <django.forms.fields.CharField object>, 'from_email': <django
declared_fields = {'body': <django.forms.fields.CharField object>, 'from_email': <dj
media
plugin_data_fields = [('from_name', ''), ('from_email', ''), ('to_name', ''), ('to_ema
```

38.4.1.1.1.713 fobi.contrib.plugins.form_handlers.mail.helpers module

```
fobi.contrib.plugins.form_handlers.mail.helpers.send_mail(subject, message, from_email,
recipient_list,
fail_silently=False,
auth_user=None,
auth_password=None,
connection=None,
html_message=None,
attachments=None)
```

Send email.

Easy wrapper for sending a single message to a recipient list. All members of the recipient list will see the other recipients in the 'To' field.

If `auth_user` is `None`, the `EMAIL_HOST_USER` setting is used. If `auth_password` is `None`, the `EMAIL_HOST_PASSWORD` setting is used.

Note: The API for this method is frozen. New code wanting to extend the functionality should use the `EmailMessage` class directly.

38.4.1.1.1.714 `fobi.contrib.plugins.form_handlers.mail.mixins` module

class `fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin`

Bases: `object`

Mail handler mixin.

get_base_url (*request*)

Get base URL.

Might be used in integration packages.

get_rendered_data (*cleaned_data, field_name_to_label_map, base_url*)

Get rendered data.

Might be used in integration packages.

send_email (*rendered_data, files*)

Send email.

Might be used in integration packages.

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

38.4.1.1.1.716 `fobi.contrib.plugins.form_handlers.mail.widgets` module

class `fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget` (*attrs=None*)

Bases: `django.forms.widgets.Textarea`

Multi email widget.

is_hidden = `False`

media

prep_value (*value*)

Prepare value before effectively render widget

render (*name, value, attrs=None, **kwargs*)

Render.

38.4.1.1.1.717 Module contents

38.4.1.1.1.718 Module contents

38.4.1.1.1.719 fobi.contrib.plugins.form_importers package

38.4.1.1.1.720 Subpackages

38.4.1.1.1.721 fobi.contrib.plugins.form_importers.mailchimp_importer package

38.4.1.1.1.722 Submodules

38.4.1.1.1.723 fobi.contrib.plugins.form_importers.mailchimp_importer.apps module

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_plugins_form_importers_mailchimp_importer'
    name = 'fobi.contrib.plugins.form_importers.mailchimp_importer'
```

38.4.1.1.1.724 fobi.contrib.plugins.form_importers.mailchimp_importer.base module

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.base.MailChimpImporter(form_entry,
                                                                    form_element,
                                                                    form_processor,
                                                                    form_data)
    Bases: fobi.form_importers.BaseFormImporter
    MailChimp data importer.
    extract_field_properties(field_data)
        Extract field properties.
        Handle choices differently as we know what the mailchimp format is.
    field_properties_mapping = {'choices': 'choices', 'help_text': 'helptext', 'initial':
    field_type_prop_name = 'field_type'
    fields_mapping = {'address': 'text', 'date': 'date', 'dropdown': 'select', 'email':
    name = <django.utils.functional.lazy.<locals>.__proxy__ object>
    position_prop_name = 'order'
    templates = ['mailchimp_importer/0.html', 'mailchimp_importer/1.html']
    uid = 'mailchimp'
    wizard
        alias of fobi.contrib.plugins.form_importers.mailchimp_importer.views.
        MailchimpImporterWizardView
```

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

class fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.**MailChimp**

Bases: *fobi.form_importers.BaseFormImporter*

MailChimp data importer.

extract_field_properties (*field_data*)

Extract field properties.

Handle choices differently as we know what the mailchimp format is.

field_properties_mapping = {'choices': 'choices', 'help_text': 'helptext', 'initial'

field_type_prop_name = 'field_type'

fields_mapping = {'address': 'text', 'date': 'date', 'dropdown': 'select', 'email':

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

position_prop_name = 'order'

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

uid = 'mailchimp'

wizard

alias of *fobi.contrib.plugins.form_importers.mailchimp_importer.views.MailchimpImporterWizardView*

38.4.1.1.1.726 fobi.contrib.plugins.form_importers.mailchimp_importer.forms module

class fobi.contrib.plugins.form_importers.mailchimp_importer.forms.**MailchimpAPIKeyForm** (*data=*

files=
auto=
pre=
fix=N
ini=
tial=
er=
ror_c
'djam
la=
bel_s
empt
field_
use_

Bases: *django.forms.forms.Form*

MailchimpAPIKeyForm.

First form the the wizard. Here users are supposed to provide the API key of their Mailchimp account.

base_fields = {'api_key': <django.forms.fields.CharField object>}

declared_fields = {'api_key': <django.forms.fields.CharField object>}

media

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpListIDForm (*arg,
                                                                    **kw)
    Bases: django.forms.forms.Form
    MailchimpListIDForm.
    Second form of the wizard. Here users are supposed to choose the form they want to import.
    base_fields = {'list_id': <django.forms.fields.ChoiceField object>}
    declared_fields = {'list_id': <django.forms.fields.ChoiceField object>}
media
```

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

```
class fobi.contrib.plugins.form_importers.mailchimp_importer.views.MailchimpImporterWizardView
    Bases: fobi.wizard.views.views.SessionWizardView
    MailchimpImporterWizardView.
    done (form_list, **kwargs)
        This method must be overridden by a subclass to process the form data after processing all steps.
    form_list = [<class 'fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailchimpListIDForm'>]
    get_form_kwargs (step)
        Get form kwargs.
```

38.4.1.1.1.728 Module contents

38.4.1.1.1.729 Module contents

38.4.1.1.1.730 Module contents

38.4.1.1.1.731 fobi.contrib.themes package

38.4.1.1.1.732 Subpackages

38.4.1.1.1.733 fobi.contrib.themes.bootstrap3 package

38.4.1.1.1.734 Subpackages

38.4.1.1.1.735 fobi.contrib.themes.bootstrap3.widgets package

38.4.1.1.1.736 Subpackages

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

38.4.1.1.1.738 Subpackages

38.4.1.1.1.739 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget package

38.4.1.1.1.740 Submodules

38.4.1.1.1.741 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.apps module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.apps.AppConfig
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_bootstrap3_widgets_form_elements_content_markdown_bootstrap3_widget'
    name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget'
```

38.4.1.1.1.742 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.fobi module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_markdown_bootstrap3_widget.fobi.widget.ContentMarkdownBootstrap3Widget
    Bases: fobi.contrib.plugins.form_elements.content.content_markdown.widgets.BaseContentMarkdownPluginWidget
    ContentMarkdownText plugin widget for Bootstrap 3.
    theme_uid = 'bootstrap3'
```

38.4.1.1.1.743 Module contents

38.4.1.1.1.744 `fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget` package

38.4.1.1.1.745 Submodules

38.4.1.1.1.746 `fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.apps` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.apps.AppConfig(
    django.apps.config.AppConfig,
    Config,
    label = 'fobi_contrib_themes_bootstrap3_widgets_form_elements_content_richtext_bootstrap3_widget',
    name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget')
```

38.4.1.1.1.747 `fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.fobi_form_elements_content_richtext_bootstrap3_widget` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.fobi_form_elements_content_richtext_bootstrap3_widget(
    fobi.contrib.plugins.form_elements.content.content_richtext.widgets.BaseContentRichTextPluginWidget,
    ContentRichText plugin widget for Bootstrap 3.
    theme_uid = 'bootstrap3')
```

38.4.1.1.1.748 Module contents

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

38.4.1.1.1.750 Submodules

38.4.1.1.1.751 `fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps.AppConfig(
    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')
```

38.4.1.1.1.752 `fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements
    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']
    theme_uid = 'bootstrap3'
```

38.4.1.1.1.753 Module contents

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

38.4.1.1.1.755 Submodules

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

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

38.4.1.1.1.757 `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_form_elements
    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']
    theme_uid = 'bootstrap3'
```

38.4.1.1.1.758 Module contents

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

38.4.1.1.1.760 Submodules

38.4.1.1.1.761 `fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps.Conf
```

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'
```

38.4.1.1.1.762 `fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_element` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_for
```

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

`BaseDummyPluginWidget`

Dummy plugin widget for Bootstrap 3.

```
media_css = []
```

```
media_js = []
```

```
theme_uid = 'bootstrap3'
```

38.4.1.1.1.763 Module contents

38.4.1.1.1.764 `fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget` package

38.4.1.1.1.765 Submodules

38.4.1.1.1.766 `fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps` module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps.C
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'fobi_contrib_themes_bootstrap3_widgets_form_elements_slider_bootstrap3_widget'
```

```
name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget'
```


38.4.1.1.1.767 fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_form_elements module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_form_elements.  
    Bases: fobi.contrib.plugins.form_elements.fields.slider.widgets.BaseSliderPluginWidget  
    Slider plugin widget for Bootstrap 3.  
    media_css = ['bootstrap3/css/bootstrap-slider.min.css', 'bootstrap3/css/fobi.plugin.slider.min.css']  
    media_js = ['bootstrap3/js/bootstrap-slider.min.js', 'bootstrap3/js/fobi.plugin.slider.min.js']  
    theme_uid = 'bootstrap3'
```

38.4.1.1.1.768 Module contents

38.4.1.1.1.769 Module contents

38.4.1.1.1.770 Module contents

38.4.1.1.1.771 Submodules

38.4.1.1.1.772 fobi.contrib.themes.bootstrap3.apps module

```
class fobi.contrib.themes.bootstrap3.apps.Config(app_name, app_module)  
    Bases: django.apps.config AppConfig  
    Config.  
    label = 'fobi_contrib_themes_bootstrap3'  
    name = 'fobi.contrib.themes.bootstrap3'
```

38.4.1.1.1.773 fobi.contrib.themes.bootstrap3.fobi_themes module

```
class fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme(user=None)  
    Bases: fobi.base.BaseTheme  
    Bootstrap3 theme.  
    add_form_element_entry_ajax_template = 'bootstrap3/add_form_element_entry_ajax.html'  
    add_form_element_entry_template = 'bootstrap3/add_form_element_entry.html'  
    add_form_handler_entry_ajax_template = 'bootstrap3/add_form_handler_entry_ajax.html'  
    add_form_handler_entry_template = 'bootstrap3/add_form_handler_entry.html'  
    add_form_wizard_handler_entry_ajax_template = 'bootstrap3/add_form_wizard_handler_entry_ajax.html'  
    add_form_wizard_handler_entry_template = 'bootstrap3/add_form_wizard_handler_entry.html'  
    base_template = 'bootstrap3/base.html'  
    create_form_entry_ajax_template = 'bootstrap3/create_form_entry_ajax.html'  
    create_form_entry_template = 'bootstrap3/create_form_entry.html'
```

```
create_form_wizard_entry_ajax_template = 'bootstrap3/create_form_wizard_entry_ajax.html'
create_form_wizard_entry_template = 'bootstrap3/create_form_wizard_entry.html'
dashboard_template = 'bootstrap3/dashboard.html'
edit_form_element_entry_ajax_template = 'bootstrap3/edit_form_element_entry_ajax.html'
edit_form_element_entry_template = 'bootstrap3/edit_form_element_entry.html'
edit_form_entry_ajax_template = 'bootstrap3/edit_form_entry_ajax.html'
edit_form_entry_template = 'bootstrap3/edit_form_entry.html'
edit_form_handler_entry_ajax_template = 'bootstrap3/edit_form_handler_entry_ajax.html'
edit_form_handler_entry_template = 'bootstrap3/edit_form_handler_entry.html'
edit_form_wizard_entry_ajax_template = 'bootstrap3/edit_form_wizard_entry_ajax.html'
edit_form_wizard_entry_template = 'bootstrap3/edit_form_wizard_entry.html'
edit_form_wizard_handler_entry_ajax_template = 'bootstrap3/edit_form_wizard_handler_entry_ajax.html'
edit_form_wizard_handler_entry_template = 'bootstrap3/edit_form_wizard_handler_entry.html'
embed_form_entry_submitted_ajax_template = 'bootstrap3/embed_form_entry_submitted_ajax.html'
form_ajax = 'bootstrap3/snippets/form_ajax.html'
form_delete_form_entry_option_class = 'glyphicon glyphicon-remove'
form_edit_form_entry_option_class = 'glyphicon glyphicon-edit'
form_element_checkbox_html_class = 'checkbox'
form_element_html_class = 'form-control'
form_entry_submitted_ajax_template = 'bootstrap3/form_entry_submitted_ajax.html'
form_entry_submitted_template = 'bootstrap3/form_entry_submitted.html'
form_importer_ajax_template = 'bootstrap3/form_importer_ajax.html'
form_importer_template = 'bootstrap3/form_importer.html'
form_list_container_class = 'list-inline'
form_non_field_and_hidden_errors_snippet_template = 'bootstrap3/snippets/form_non_field_and_hidden_errors.html'
form_properties_snippet_template_name = 'bootstrap3/snippets/form_properties_snippet.html'
form_snippet_template_name = 'bootstrap3/snippets/form_snippet.html'
form_view_form_entry_option_class = 'glyphicon glyphicon-list'
form_wizard_ajax = 'bootstrap3/snippets/form_wizard_ajax.html'
form_wizard_properties_snippet_template_name = 'bootstrap3/snippets/form_wizard_properties_snippet.html'
form_wizard_snippet_template_name = 'bootstrap3/snippets/form_wizard_snippet.html'
form_wizard_template = 'bootstrap3/snippets/form_wizard.html'
form_wizards_dashboard_template = 'bootstrap3/form_wizards_dashboard.html'
forms_list_template = 'bootstrap3/forms_list.html'
master_base_template = 'bootstrap3/_base.html'
media_css = ('bootstrap3/css/bootstrap.css', 'bootstrap3/css/bootstrap3_fobi_extras.css')
```

```
media_js = ('js/jquery-1.10.2.min.js', 'jquery-ui/js/jquery-ui-1.10.4.custom.min.js',
messages_snippet_template_name = 'bootstrap3/snippets/messages_snippet.html'
name = <django.utils.functional.lazy.<locals>.__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'
view_form_wizard_entry_ajax_template = 'bootstrap3/view_form_wizard_entry_ajax.html'
view_form_wizard_entry_template = 'bootstrap3/view_form_wizard_entry.html'
```

38.4.1.1.1.774 Module contents

38.4.1.1.1.775 fobi.contrib.themes.djangocms_admin_style_theme package

38.4.1.1.1.776 Subpackages

38.4.1.1.1.777 fobi.contrib.themes.djangocms_admin_style_theme.widgets package

38.4.1.1.1.778 Subpackages

38.4.1.1.1.779 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements package

38.4.1.1.1.780 Subpackages

38.4.1.1.1.781 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin package

38.4.1.1.1.782 Submodules

38.4.1.1.1.783 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin.AdminForm(
    Bases: django.apps.config.AppConfig
    Config.
    label = 'fobi_contrib_themes.djangocms_admin_style_theme_widgets_form_elements_content_richtext_admin'
    name = 'fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin'
```

38.4.1.1.1.784 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin.  
    Bases: fobi.contrib.plugins.form_elements.content.content_richtext.widgets.  
            BaseContentRichTextPluginWidget  
    ContentRichText plugin widget for.djangocms_admin_style theme.  
    theme_uid = 'simple'
```

38.4.1.1.1.785 Module contents

38.4.1.1.1.786 Module contents

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

38.4.1.1.1.788 Subpackages

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

38.4.1.1.1.790 Submodules

38.4.1.1.1.791 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps.AppConfig  
    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.apps'
```

38.4.1.1.1.792 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements module

```
class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements.  
    Bases: fobi.contrib.plugins.form_handlers.db_store.widgets.  
            BaseDbStorePluginWidget  
    DbStore plugin widget for.djangocms_admin_style_theme theme.  
    export_entries_icon_class = ''  
    theme_uid = 'djangocms_admin_style_theme'  
    view_entries_icon_class = ''
```

38.4.1.1.1.793 Module contents

38.4.1.1.1.794 Module contents

38.4.1.1.1.795 Module contents

38.4.1.1.1.796 Submodules

38.4.1.1.1.797 fobi.contrib.themes.djangocms_admin_style_theme.apps module

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

38.4.1.1.1.798 fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes module

```
class fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme
    Bases: fobi.base.BaseTheme
    A theme that has a native djangocms-admin-style style.
    add_form_element_entry_ajax_template = 'djangocms_admin_style_theme/add_form_element_entry_ajax.html'
    add_form_element_entry_template = 'djangocms_admin_style_theme/add_form_element_entry.html'
    add_form_handler_entry_ajax_template = 'djangocms_admin_style_theme/add_form_handler_entry_ajax.html'
    add_form_handler_entry_template = 'djangocms_admin_style_theme/add_form_handler_entry.html'
    base_edit_template = 'djangocms_admin_style_theme/base_edit.html'
    base_template = 'djangocms_admin_style_theme/base.html'
    base_view_template = 'djangocms_admin_style_theme/base_view.html'
    create_form_entry_ajax_template = 'djangocms_admin_style_theme/create_form_entry_ajax.html'
    create_form_entry_template = 'djangocms_admin_style_theme/create_form_entry.html'
    dashboard_template = 'djangocms_admin_style_theme/dashboard.html'
    edit_form_element_entry_ajax_template = 'djangocms_admin_style_theme/edit_form_element_entry_ajax.html'
    edit_form_element_entry_template = 'djangocms_admin_style_theme/edit_form_element_entry.html'
    edit_form_entry_ajax_template = 'djangocms_admin_style_theme/edit_form_entry_ajax.html'
    classmethod edit_form_entry_edit_option_html()
        For adding the edit link to edit form entry view.
        Return str
    classmethod edit_form_entry_help_text_extra()
        For adding the edit link to edit form entry view.
        Return str
```

```
edit_form_entry_template = 'djangocms_admin_style_theme/edit_form_entry.html'
edit_form_handler_entry_ajax_template = 'djangocms_admin_style_theme/edit_form_handler_
edit_form_handler_entry_template = 'djangocms_admin_style_theme/edit_form_handler_entr
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_snip
form_element_checkbox_html_class = 'checkbox'
form_element_html_class = 'vTextField'
form_entry_submitted_ajax_template = 'djangocms_admin_style_theme/form_entry_submitted
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_pro
form_radio_element_html_class = 'radiolist'
form_snippet_template_name = 'djangocms_admin_style_theme/snippets/form_snippet.html'
form_view_form_entry_option_class = 'viewlink'
form_view_snippet_template_name = 'djangocms_admin_style_theme/snippets/form_view_snip
forms_list_template = 'djangocms_admin_style_theme/forms_list.html'
import_form_entry_ajax_template = 'djangocms_admin_style_theme/import_form_entry_ajax.l
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', '
media_js = ('js/jquery-1.10.2.min.js', 'jquery-ui/js/jquery-ui-1.10.4.custom.min.js',
messages_snippet_template_name = 'djangocms_admin_style_theme/snippets/messages_snippe
name = <django.utils.functional.lazy.<locals>.__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'
```

38.4.1.1.1.799 Module contents

38.4.1.1.1.800 fobi.contrib.themes.foundation5 package

38.4.1.1.1.801 Subpackages

38.4.1.1.1.802 fobi.contrib.themes.foundation5.widgets package

38.4.1.1.1.803 Subpackages

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

38.4.1.1.1.805 Subpackages

38.4.1.1.1.806 fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget package

38.4.1.1.1.807 Submodules

38.4.1.1.1.808 fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget.apps module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_w
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_foundation5_widgets_form_elements_content_richtext_founda
    name = 'fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundat
```

38.4.1.1.1.809 fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_widget.fobi module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext_foundation5_w
    Bases: fobi.contrib.plugins.form_elements.content.content_richtext.widgets.
            BaseContentRichTextPluginWidget
    Content rich-text plugin widget for Foundation 5.
    theme_uid = 'foundation5'
```

38.4.1.1.1.810 Module contents

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

38.4.1.1.1.812 Submodules

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

```
class fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps.C
    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'
```

38.4.1.1.1.814 `fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_form_element` module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.fobi_f
    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', 'f
    theme_uid = 'foundation5'
```

38.4.1.1.1.815 Module contents

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

38.4.1.1.1.817 Submodules

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

```
class fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.ap
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_foundation5_widgets_form_elements_datetime_foundation5_wi
    name = 'fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_wid
```


38.4.1.1.1.819 fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.fobi_form_el 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']
    theme_uid = 'foundation5'
```

38.4.1.1.1.820 Module contents

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

38.4.1.1.1.822 Submodules

38.4.1.1.1.823 fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps.C
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_foundation5_widgets_form_elements_dummy_foundation5_widge
    name = 'fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget'
```

38.4.1.1.1.824 fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_el module

```
class fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_
    Bases: fobi.contrib.plugins.form_elements.test.dummy.widgets.
            BaseDummyPluginWidget
    Dummy plugin widget for Foundation 5.
    media_css = []
    media_js = []
    theme_uid = 'foundation5'
```

38.4.1.1.1.825 Module contents

38.4.1.1.1.826 Module contents

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

38.4.1.1.1.828 Subpackages

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

38.4.1.1.1.830 Submodules

38.4.1.1.1.831 fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps module

```
class fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps.AppConfig
    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'
```

38.4.1.1.1.832 fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_element module

```
class fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_element
    Bases: fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget
    DbStore plugin widget for Foundation 5.
    export_entries_icon_class = 'fi-page-export'
    theme_uid = 'foundation5'
    view_entries_icon_class = 'fi-list'
    view_saved_form_data_entries_template_name = 'db_store_foundation5_widget/view_saved_form_data.html'
```

38.4.1.1.1.833 Module contents

38.4.1.1.1.834 Module contents

38.4.1.1.1.835 Module contents

38.4.1.1.1.836 Submodules

38.4.1.1.1.837 fobi.contrib.themes.foundation5.apps module

```
class fobi.contrib.themes.foundation5.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_foundation5'
    name = 'fobi.contrib.themes.foundation5'
```

38.4.1.1.1.838 fobi.contrib.themes.foundation5.fobi_themes module

```
class fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme(user=None)
    Bases: fobi.base.BaseTheme
```

Foundation5 theme.

Based on the “Workspace” example of the Foundation 5. Click [here](#) for more.

```
add_form_element_entry_ajax_template = 'foundation5/add_form_element_entry_ajax.html'
add_form_element_entry_template = 'foundation5/add_form_element_entry.html'
add_form_handler_entry_ajax_template = 'foundation5/add_form_handler_entry_ajax.html'
add_form_handler_entry_template = 'foundation5/add_form_handler_entry.html'
base_template = 'foundation5/base.html'
create_form_entry_ajax_template = 'foundation5/create_form_entry_ajax.html'
create_form_entry_template = 'foundation5/create_form_entry.html'
create_form_wizard_entry_ajax_template = 'foundation5/create_form_wizard_entry_ajax.html'
create_form_wizard_entry_template = 'foundation5/create_form_wizard_entry.html'
dashboard_template = 'foundation5/dashboard.html'
edit_form_element_entry_ajax_template = 'foundation5/edit_form_element_entry_ajax.html'
edit_form_element_entry_template = 'foundation5/edit_form_element_entry.html'
edit_form_entry_ajax_template = 'foundation5/edit_form_entry_ajax.html'
edit_form_entry_template = 'foundation5/edit_form_entry.html'
edit_form_handler_entry_ajax_template = 'foundation5/edit_form_handler_entry_ajax.html'
edit_form_handler_entry_template = 'foundation5/edit_form_handler_entry.html'
edit_form_wizard_entry_ajax_template = 'foundation5/edit_form_wizard_entry_ajax.html'
edit_form_wizard_entry_template = 'foundation5/edit_form_wizard_entry.html'
```

```
form_ajax = 'foundation5/snippets/form_ajax.html'
form_delete_form_entry_option_class = 'fi-page-delete'
form_edit_form_entry_option_class = 'fi-page-edit'
form_element_checkbox_html_class = 'checkbox'
form_element_html_class = 'form-control'
form_entry_submitted_ajax_template = 'foundation5/form_entry_submitted_ajax.html'
form_entry_submitted_template = 'foundation5/form_entry_submitted.html'
form_list_container_class = 'inline-list'
form_non_field_and_hidden_errors_snippet_template = 'foundation5/snippets/form_non_fie
form_properties_snippet_template_name = 'foundation5/snippets/form_properties_snippet.l
form_snippet_template_name = 'foundation5/snippets/form_snippet.html'
form_view_form_entry_option_class = 'fi-list'
form_wizard_ajax = 'foundation5/snippets/form_wizard_ajax.html'
form_wizard_properties_snippet_template_name = 'foundation5/snippets/form_wizard_prope
form_wizard_snippet_template_name = 'foundation5/snippets/form_wizard_snippet.html'
form_wizards_dashboard_template = 'foundation5/form_wizards_dashboard.html'
forms_list_template = 'foundation5/forms_list.html'
import_form_entry_ajax_template = 'foundation5/import_form_entry_ajax.html'
import_form_entry_template = 'foundation5/import_form_entry.html'
master_base_template = 'foundation5/_base.html'
media_css = ('foundation5/css/foundation.min.css', 'foundation5/css/foundation_fobi_ex
media_js = ('foundation5/js/vendor/modernizr.js', 'foundation5/js/vendor/jquery.js', '
messages_snippet_template_name = 'foundation5/snippets/messages_snippet.html'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'foundation5'
view_form_entry_ajax_template = 'foundation5/view_form_entry_ajax.html'
view_form_entry_template = 'foundation5/view_form_entry.html'
view_form_wizard_entry_ajax_template = 'foundation5/view_form_wizard_entry_ajax.html'
view_form_wizard_entry_template = 'foundation5/view_form_wizard_entry.html'
```

38.4.1.1.1.839 Module contents

38.4.1.1.1.840 fobi.contrib.themes.simple package

38.4.1.1.1.841 Subpackages

38.4.1.1.1.842 fobi.contrib.themes.simple.widgets package

38.4.1.1.1.843 Subpackages

38.4.1.1.1.844 fobi.contrib.themes.simple.widgets.form_elements package

38.4.1.1.1.845 Subpackages

38.4.1.1.1.846 fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget package

38.4.1.1.1.847 Submodules

38.4.1.1.1.848 fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.apps module

```
class fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.apps
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_simple_widgets_form_elements_content_richtext_simple_widget'
    name = 'fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget'
```

38.4.1.1.1.849 fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi_form_elements module

```
class fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi_form_elements
    Bases: fobi.contrib.plugins.form_elements.content.content_richtext.widgets.BaseContentRichTextPluginWidget
    ContentRichText plugin widget for Simple theme.
    theme_uid = 'simple'
```

38.4.1.1.1.850 Module contents

38.4.1.1.1.851 Module contents

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

38.4.1.1.1.853 Subpackages

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

38.4.1.1.1.855 Submodules

38.4.1.1.1.856 fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps module

```
class fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    Config.
    label = 'fobi_contrib_themes_simple_widgets_form_handlers_db_store'
    name = 'fobi.contrib.themes.simple.widgets.form_handlers.db_store'
```

38.4.1.1.1.857 fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements module

```
class fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements.DbStore
    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 = ''
```

38.4.1.1.1.858 Module contents

38.4.1.1.1.859 Module contents

38.4.1.1.1.860 Module contents

38.4.1.1.1.861 Submodules

38.4.1.1.1.862 fobi.contrib.themes.simple.apps module

```
class fobi.contrib.themes.simple.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    Config.
```

```
label = 'fobi_contrib_themes_simple'
name = 'fobi.contrib.themes.simple'
```

38.4.1.1.1.863 fobi.contrib.themes.simple.fobi_themes module

```
class fobi.contrib.themes.simple.fobi_themes.SimpleTheme (user=None)
    Bases: fobi.base.BaseTheme
    Simple theme that has a native Django style.

    add_form_element_entry_ajax_template = 'simple/add_form_element_entry_ajax.html'
    add_form_element_entry_template = 'simple/add_form_element_entry.html'
    add_form_handler_entry_ajax_template = 'simple/add_form_handler_entry_ajax.html'
    add_form_handler_entry_template = 'simple/add_form_handler_entry.html'
    base_edit_template = 'simple/base_edit.html'
    base_template = 'simple/base.html'
    base_view_template = 'simple/base_view.html'
    create_form_entry_ajax_template = 'simple/create_form_entry_ajax.html'
    create_form_entry_template = 'simple/create_form_entry.html'
    create_form_wizard_entry_ajax_template = 'simple/create_form_wizard_entry_ajax.html'
    create_form_wizard_entry_template = 'simple/create_form_wizard_entry.html'
    dashboard_template = 'simple/dashboard.html'
    edit_form_element_entry_ajax_template = 'simple/edit_form_element_entry_ajax.html'
    edit_form_element_entry_template = 'simple/edit_form_element_entry.html'
    edit_form_entry_ajax_template = 'simple/edit_form_entry_ajax.html'
    edit_form_entry_template = 'simple/edit_form_entry.html'
    edit_form_handler_entry_ajax_template = 'simple/edit_form_handler_entry_ajax.html'
    edit_form_handler_entry_template = 'simple/edit_form_handler_entry.html'
    edit_form_wizard_entry_ajax_template = 'simple/edit_form_wizard_entry_ajax.html'
    edit_form_wizard_entry_template = 'simple/edit_form_wizard_entry.html'
    form_ajax = 'simple/snippets/form_ajax.html'
    form_delete_form_entry_option_class = 'glyphicon glyphicon-remove'
    form_edit_ajax = 'simple/snippets/form_edit_ajax.html'
    form_edit_form_entry_option_class = 'glyphicon glyphicon-edit'
    form_edit_snippet_template_name = 'simple/snippets/form_edit_snippet.html'
    form_element_checkbox_html_class = 'checkbox'
    form_element_html_class = 'vTextField'
    form_entry_submitted_ajax_template = 'simple/form_entry_submitted_ajax.html'
    form_entry_submitted_template = 'simple/form_entry_submitted.html'
```

```
form_list_container_class = 'list-inline'
form_properties_snippet_template_name = 'simple/snippets/form_properties_snippet.html'
form_radio_element_html_class = 'radiolist'
form_snippet_template_name = 'simple/snippets/form_snippet.html'
form_view_form_entry_option_class = 'glyphicon glyphicon-list'
form_view_snippet_template_name = 'simple/snippets/form_view_snippet.html'
form_wizard_properties_snippet_template_name = 'simple/snippets/form_wizard_properties'
form_wizard_snippet_template_name = 'simple/snippets/form_wizard_snippet.html'
forms_list_template = 'simple/forms_list.html'
import_form_entry_ajax_template = 'simple/import_form_entry_ajax.html'
import_form_entry_template = 'simple/import_form_entry.html'
master_base_template = 'simple/_base.html'
media_css = ('simple/css/fobi.simple.css', 'jquery-ui/css/django-admin-theme/jquery-ui'
media_js = ('js/jquery-1.10.2.min.js', 'jquery-ui/js/jquery-ui-1.10.4.custom.min.js',
messages_snippet_template_name = 'simple/snippets/messages_snippet.html'
name = <django.utils.functional.lazy.<locals>.__proxy__ object>
uid = 'simple'
view_form_entry_ajax_template = 'simple/view_form_entry_ajax.html'
view_form_entry_template = 'simple/view_form_entry.html'
view_form_wizard_entry_ajax_template = 'simple/view_form_wizard_entry_ajax.html'
view_form_wizard_entry_template = 'simple/view_form_wizard_entry.html'
```

38.4.1.1.1.864 Module contents

38.4.1.1.1.865 Module contents

38.4.1.1.2 Module contents

38.4.1.2 fobi.integration package

38.4.1.2.1 Submodules

38.4.1.2.2 fobi.integration.helpers module

`fobi.integration.helpers.get_template_choices` (*source*, *choices*,
theme_specific_choices_key)

Get the template choices.

It's possible to provide theme templates per theme or just per project.

Parameters

- **source** (*str*) – Example value 'feincms_integration'.

- **or list choices** (*tuple*) –
- **theme_specific_choices_key** (*str*) –

Return list

38.4.1.2.3 fobi.integration.processors module

class `fobi.integration.processors.IntegrationProcessor`

Bases: `object`

Generic integration processor.

Parameters

- **form_sent_get_param** (*str*) –
- **can_redirect** (*bool*) – If set to True, if not authenticated an attempt to redirect user to a login page would be made. Otherwise, a message about authentication would be generated instead (in place of the form). Some content management systems, like Django-CMS, aren't able to redirect on plugin level. For those systems, the value of `can_redirect` should be set to False.
- **login_required_template_name** (*str*) – Template to be used for rendering the login required message. This is only important when `login_required_redirect` is set to False.

`can_redirect = True`

`form_sent_get_param = 'sent'`

`get_context_data` (*request, instance, **kwargs*)
Get context data.

`get_form_template_name` (*request, instance*)
Get form template name.

`get_login_required_template_name` (*request, instance*)
Get login required template name.

`get_process_form_redirect_url` (*request, instance*)
Get process form redirect URL (success).

Parameters

- **request** (*django.http.HttpRequest*) –
- **instance** (*fobi.models.FormEntry*) –

Return str

`get_success_page_template_name` (*request, instance*)
Get success page template name.

`integration_check` (*instance*)
Integration check.

Performs a simple check to identify whether the model instance has been implemented according to the expectations.

`login_required_template_name = 'fobi/integration/login_required.html'`

38.4.1.2.4 Module contents

38.4.1.3 fobi.management package

38.4.1.3.1 Subpackages

38.4.1.3.1.1 fobi.management.commands package

38.4.1.3.1.2 Submodules

38.4.1.3.1.3 fobi.management.commands.fobi_find_broken_entries module

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

    Bases: django.core.management.base.BaseCommand

    Find the broken plugin records in the database:

    • fobi.models.FormElementEntry
    • fobi.models.FormHandlerEntry

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

38.4.1.3.1.4 fobi.management.commands.fobi_sync_plugins module

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

    Bases: django.core.management.base.BaseCommand

    Adds the missing plugins to database.

    This command shall be ran every time a developer adds a new plugin. The following plugins are affected:

    • fobi.models.FormElementPlugin
    • fobi.models.FormHandlerPlugin

    add_arguments (parser)
        Entry point for subclassed commands to add custom arguments.

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

38.4.1.3.1.5 fobi.management.commands.fobi_update_plugin_data module

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

    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.
```

38.4.1.3.1.6 Module contents

38.4.1.3.2 Module contents

38.4.1.4 fobi.migrations package

38.4.1.4.1 Submodules

38.4.1.4.2 fobi.migrations.0001_initial module

```
class fobi.migrations.0001_initial.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('auth', '__first__'), ('auth', '0006_require_contenttypes_0002')]
    operations = [<CreateModel name='FormElement', fields=[('id', <django.db.models.fields
```

38.4.1.4.3 fobi.migrations.0002_auto_20150912_1744 module

```
class fobi.migrations.0002_auto_20150912_1744.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0001_initial')]
    operations = [<AddField model_name='formentry', name='created', field=<django.db.model
```

38.4.1.4.4 fobi.migrations.0003_auto_20160517_1005 module

```
class fobi.migrations.0003_auto_20160517_1005.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0002_auto_20150912_1744')]
    operations = [<AlterField model_name='formelement', name='plugin_uid', field=<django.d
```

38.4.1.4.5 fobi.migrations.0004_auto_20160906_1513 module

```
class fobi.migrations.0004_auto_20160906_1513.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0003_auto_20160517_1005')]
    operations = [<AlterField model_name='formelement', name='plugin_uid', field=<django.d
```

38.4.1.4.6 fobi.migrations.0005_auto_20160908_1457 module

```
class fobi.migrations.0005_auto_20160908_1457.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0004_auto_20160906_1513')]

    operations = [<AlterField model_name='formelement', name='plugin_uid', field=<django.db.models.fields.CharField object>]
```

38.4.1.4.7 fobi.migrations.0006_auto_20160911_1549 module

```
class fobi.migrations.0006_auto_20160911_1549.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0005_auto_20160908_1457')]

    operations = [<AlterField model_name='formhandler', name='plugin_uid', field=<django.db.models.fields.CharField object>]
```

38.4.1.4.8 fobi.migrations.0007_auto_20160926_1652 module

```
class fobi.migrations.0007_auto_20160926_1652.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0006_auto_20160911_1549')]

    operations = [<CreateModel name='FormWizardFormEntry', fields=[('id', <django.db.models.AutoField object>), ('form_handler', <django.db.models.ForeignKey object>), ('wizard_type', <django.db.models.ForeignKey object>)]]
```

38.4.1.4.9 fobi.migrations.0008_formwizardhandlerentry module

```
class fobi.migrations.0008_formwizardhandlerentry.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0007_auto_20160926_1652')]

    operations = [<CreateModel name='FormWizardHandlerEntry', fields=[('id', <django.db.models.AutoField object>), ('form_wizard_entry', <django.db.models.ForeignKey object>)]]
```

38.4.1.4.10 fobi.migrations.0009_formwizardentry_wizard_type module

```
class fobi.migrations.0009_formwizardentry_wizard_type.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0008_formwizardhandlerentry')]

    operations = [<AddField model_name='formwizardentry', name='wizard_type', field=<django.db.models.ForeignKey object>]
```

38.4.1.4.11 fobi.migrations.0010_formwizardhandler module

```
class fobi.migrations.0010_formwizardhandler.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('auth', '0007_alter_validators_add_error_messages'), ('auth', '__first__')]

    operations = [<CreateModel name='FormWizardHandler', fields=[('id', <django.db.models.AutoField object>), ('form_wizard_handler_entry', <django.db.models.ForeignKey object>)]]
```

38.4.1.4.12 fobi.migrations.0011_formentry_title module

```
class fobi.migrations.0011_formentry_title.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0010_formwizardhandler')]

    operations = [<AddField model_name='formentry', name='title', field=<django.db.models.
```

38.4.1.4.13 fobi.migrations.0012_auto_20161109_1550 module

```
class fobi.migrations.0012_auto_20161109_1550.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0011_formentry_title')]

    operations = [<AddField model_name='formwizardentry', name='title', field=<django.db.m
```

38.4.1.4.14 fobi.migrations.0013_formwizardentry_show_all_navigation_buttons module

```
class fobi.migrations.0013_formwizardentry_show_all_navigation_buttons.Migration(name,
                                                                                   app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0012_auto_20161109_1550')]

    operations = [<AddField model_name='formwizardentry', name='show_all_navigation_button
```

38.4.1.4.15 fobi.migrations.0014_auto_20170516_1413 module

```
class fobi.migrations.0014_auto_20170516_1413.Migration(name, app_label)
    Bases: django.db.migrations.migration.Migration

    dependencies = [('fobi', '0013_formwizardentry_show_all_navigation_buttons')]

    operations = [<AlterField model_name='formentry', name='slug', field=<autoslug.fields.
```

38.4.1.4.16 Module contents

38.4.1.5 fobi.reusable package

38.4.1.5.1 Subpackages

38.4.1.5.1.1 fobi.reusable.markdown_widget package

38.4.1.5.1.2 Submodules

38.4.1.5.1.3 fobi.reusable.markdown_widget.helpers module

```
fobi.reusable.markdown_widget.helpers.convert_to_markdown(content)
    Trans-compiles Markdown text to HTML.
```

Parameters **content** (*str*) – Markdown text.

Returns HTML encoded text.

Return type str

38.4.1.5.1.4 fobi.reusable.markdown_widget.widgets module

class fobi.reusable.markdown_widget.widgets.**MarkdownWidget** (*attrs=None*)

Bases: django.forms.widgets.Textarea

Markdown widget based on remarkable.

class **Media**

Bases: object

Media options.

js = ['markdown_widget/remarkable.min.js', 'content_markdown/fobi.plugin.content_ma

media

render (*name, value, attrs=None, **kwargs*)

Returns this Widget rendered as HTML, as a Unicode string.

The ‘value’ given is not guaranteed to be valid input, so subclass implementations should program defensively.

38.4.1.5.1.5 Module contents

38.4.1.5.2 Module contents

38.4.1.6 fobi.templatetags package

38.4.1.6.1 Submodules

38.4.1.6.2 fobi.templatetags.fobi_tags module

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 [con-
text_var_name] %}
```

Example

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

fobi.templatetags.fobi_tags.**get_fobi_form_wizard_handler_plugin_custom_actions** (*parser,*
to-
ken)

Get the form wizard handler plugin custom actions.

Note, that plugin shall be a instance of fobi.models.FormWizardHandlerEntry.

Syntax

```
{% get_fobi_form_wizard_handler_plugin_custom_actions [plugin] [form_wizard_entry]
as [context_var_name] %}
```

Example

```
{% get_fobi_form_wizard_handler_plugin_custom_actions
    plugin form_wizard_entry as form_wizard_handler_plugin_custom_actions %}
```

`fobi.templatetags.fobi_tags.get_fobi_plugin(parser, token)`

Get the plugin.

Note, that entry shall be a instance of `fobi.models.FormElementEntry` or `fobi.models.FormHandlerEntry`.

Syntax `{% get_fobi_plugin entry as [context_var_name] %}`

Example `{% get_fobi_plugin entry as plugin %}`

```
{% get_fobi_plugin entry as plugin %} {{ plugin.render }}
```

`fobi.templatetags.fobi_tags.get_form_field_type(parser, token)`

Get form field type.

Syntax:

```
{% get_form_field_type [field] as [context_var_name] %}
```

Example:

```
{% get_form_field_type form.field as form_field_type %}
{% if form_field_type.is_checkbox %}
    ...
{% endif %}
```

`fobi.templatetags.fobi_tags.get_form_hidden_fields_errors(parser, token)`

Get form hidden fields errors.

Syntax `{% get_form_hidden_fields_errors [form] as [context_var_name] %}`

Example `{% get_form_hidden_fields_errors form as form_hidden_fields_errors %} {{
 form_hidden_fields_errors.as_ul }}`

`fobi.templatetags.fobi_tags.has_edit_form_entry_permissions(parser, token)`

Checks the permissions

Syntax `{% has_edit_form_entry_permissions as [var_name] %}`

Example `{% has_edit_form_entry_permissions %}`

or

```
{% has_edit_form_entry_permissions as has_permissions %}
```

`fobi.templatetags.fobi_tags.render_auth_link(context)`

Render auth link.

`fobi.templatetags.fobi_tags.render_fobi_forms_list(context, queryset, *args,
**kwargs)`

Render the list of fobi forms.

Syntax `{% render_fobi_forms_list [queryset] [show_edit_link] [show_delete_link]
[show_export_link] %}`

Example `{% render_fobi_forms_list queryset show_edit_link=True show_delete_link=False show_export_link=False %}`

38.4.1.6.3 fobi.templatetags.future_compat module

`fobi.templatetags.future_compat.firstof` (*parser, token*)

Outputs the first variable passed that is not False.

Outputs nothing if all the passed variables are False.

Sample usage:

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

This is equivalent to:

```
{% if var1 %}
  {{ var1 }}
{% elif var2 %}
  {{ var2 }}
{% elif var3 %}
  {{ var3 }}
{% endif %}
```

but obviously much cleaner!

You can also use a literal string as a fallback value in case all passed variables are False:

```
{% firstof var1 var2 var3 "fallback value" %}
```

If you want to disable auto-escaping of variables you can use:

```
{% autoescape off %}
  {% firstof var1 var2 var3 "<strong>fallback value</strong>" %}
{% autoescape %}
```

Or if only some variables should be escaped, you can use:

```
{% firstof var1 var2|safe var3 "<strong>fallback value</strong>"|safe %}
```

38.4.1.6.4 Module contents

38.4.1.7 fobi.tests package

38.4.1.7.1 Submodules

38.4.1.7.2 fobi.tests.base module

class `fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest` (*methodName='runTest'*)

Bases: `django.contrib.staticfiles.testing.StaticLiveServerTestCase`

Browser tests django-fobi bulding forms functionality.

Backed up by selenium. This test is based on the bootstrap3 theme.

LIVE_SERVER_URL = None


```

cleans_up_after_itself = True

reset_sequences = True

classmethod setUpClass()
    Set up class.

tearDown()
    Tear down.

classmethod tearDownClass()
    Tear down class.

```

38.4.1.7.3 fobi.tests.constants module

38.4.1.7.4 fobi.tests.core module

```

fobi.tests.core.print_info(func)
    Prints some useful info.

fobi.tests.core.skip(func)
    Simply skips the test.

fobi.tests.core.is_app_setup_completed()
    Is fobi setup completed?

fobi.tests.core.mark_app_setup_as_completed()
    Mark fobi setup as completed.

```

38.4.1.7.5 fobi.tests.data module

38.4.1.7.6 fobi.tests.helpers module

```

fobi.tests.helpers.create_form_with_entries(user=None,
                                             create_entries_if_form_exist=True,
                                             data={},
                                             is_public=False,
                                             name='Test form',
                                             slug='test-form')

```

Create test form with entries.

Fills the form with pre-defined plugins.

Parameters

- **user** (*django.contrib.auth.models.User*) –
- **create_entries_if_form_exist** (*bool*) – If set to True, entries are being created even if form already exists (a database record).
- **data** (*dict*) –
- **is_public** (*bool*) –
- **name** (*str*) –
- **slug** (*str*) –

Return **fobi.models.FormEntry** Instance of `fobi.models.FormEntry` with a number of form elements and handlers filled in.

`fobi.tests.helpers.db_clean_up` (*clean_form=False*, *clean_elements=True*,
clean_handlers=True)
 Clean up the database.
 Clean up the database by removing all form element and form handler entries.

`fobi.tests.helpers.get_or_create_admin_user` ()
 Create a user for testing the fobi.
 TODO: At the moment an admin account is being tested. Automated tests with diverse accounts are to be implemented.

`fobi.tests.helpers.get_or_create_admin_user` ()
 Create a user for testing the fobi.
 TODO: At the moment an admin account is being tested. Automated tests with diverse accounts are to be implemented.

`fobi.tests.helpers.phantom_js_clean_up` ()
 Clean up Phantom JS.
 Kills all phantomjs instances, disregard of their origin.

`fobi.tests.helpers.setup_app` (*collectstatic=False*, *fobi_sync_plugins=False*)
 Set up fobi.

38.4.1.7.7 fobi.tests.test_browser_build_dynamic_forms module

class `fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuldDynamicFormsTest` (*methodName*
Bases: fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest)
 Browser tests django-fobi building forms functionality.
 Backed up by selenium. This test is based on the bootstrap3 theme.

test_1001_open_dashboard (**args*, ***kwargs*)
 Inner.

test_2001_add_form (**args*, ***kwargs*)
 Inner.

test_2002_edit_form (**args*, ***kwargs*)
 Inner.

test_2003_delete_form (**args*, ***kwargs*)
 Inner.

test_2004_submit_form (**args*, ***kwargs*)
 Inner.

test_3001_add_form_elements (**args*, ***kwargs*)
 Inner.

test_3002_remove_form_elements (**args*, ***kwargs*)
 Inner.

test_3003_edit_form_elements (**args*, ***kwargs*)
 Inner.

test_4001_add_form_handlers (**args*, ***kwargs*)
 Inner.

```
test_4002_remove_form_handlers (*args, **kwargs)
    Inner.

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

38.4.1.7.8 fobi.tests.test_core module

```
class fobi.tests.test_core.FobiCoreTest (methodName='runTest')
    Bases: django.test.testcases.TestCase
    Tests of django-fobi core functionality.

    setUp ()
        Set up.

    test_01_get_registered_form_element_plugins (*args, **kwargs)
        Inner.

    test_02_get_registered_form_handler_plugins (*args, **kwargs)
        Inner.

    test_03_get_registered_form_callbacks (*args, **kwargs)
        Inner.

    test_04_get_registered_themes (*args, **kwargs)
        Inner.

    test_05_action_url (*args, **kwargs)
        Inner.

    test_06_form_entry_get_absolute_url (*args, **kwargs)
        Inner.

    test_07_form_wizard_entry_get_absolute_url (*args, **kwargs)
        Inner.

    test_08_form_entry_is_active (*args, **kwargs)
        Inner.
```

38.4.1.7.9 fobi.tests.test_drf_integration module

```
class fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTests (methodName='runTest')
    Bases: rest_framework.test.APITestCase
    DRF integration tests.

    setUp ()
        Set up.

    classmethod setUpTestData ()
        Load initial data for the TestCase

    tearDown ()
        Set up.

    test_01_options_action_public_form ()
        Test OPTIONS action call for public form.
```

```
test_02_put_action_public_form()
    Test PUT action call for public form.

test_03_fail_put_action_public_form()
    Test PUT action call fail test for public form.

test_04_fail_options_action_non_public_form()
    Test OPTIONS action call fail test for non-public form.

test_05_fail_put_action()
    Test PUT action call fail test for non-public form.

test_06_options_action_non_public_form_auth_user()
    Test OPTIONS action call for authorised user for non-public form.

test_07_put_action_non_public_form_auth_user()
    Test PUT action call for authorised user for non-public form.

test_08_get_action_public_form()
    Test OPTIONS action call for public form.

test_09_get_action_non_public_form_auth_user()
    Test GET action call for authorised user for non-public form.

test_10_fail_get_action_non_public_form()
    Test GET action call fail test for non-public form.
```

38.4.1.7.10 fobi.tests.test_dynamic_forms module

```
class fobi.tests.test_dynamic_forms.FobiDynamicFormsTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    Tests of django-fob dynamic forms functionality.

    setUp()
        Set up.

    test_01_assemble_form_class_and_render_form(*args, **kwargs)
        Inner.
```

38.4.1.7.11 fobi.tests.test_feincms_integration module

```
class fobi.tests.test_feincms_integration.FeinCMSIntegrationTest (methodName='runTest')
    Bases: fobi.tests.base.BaseFobiBrowserBulDynamicFormsTest

    FeinCMS integration tests.

    setUp()
        Hook method for setting up the test fixture before exercising it.

    test_fobi_form_widget_public_form()
        Test fobi form widget.
```

38.4.1.7.12 fobi.tests.test_form_importers_mailchimp module

```
class fobi.tests.test_form_importers_mailchimp.FormImportersMailchimpTest (methodName='runTest')
    Bases: django.test.testcases.TestCase

    Tests of form importers mailchimp functionality.
```

```

setUp()
    Set up.

test_01_test_mailchimp_importer(*args, **kwargs)
    Inner.

```

38.4.1.7.13 fobi.tests.test_sortable_dict module

```

class fobi.tests.test_sortable_dict.FobiDataStructuresTest(methodName='runTest')
    Bases: django.test.testcases.TestCase

    Tests of django-fobi data_structures module functionality.

    setUp()
        Set up.

    test_01_sortable_dict_move_before_key(*args, **kwargs)
        Inner.

    test_02_sortable_dict_move_after_key(*args, **kwargs)
        Inner.

```

38.4.1.7.14 Module contents

38.4.1.8 fobi.urls package

38.4.1.8.1 Submodules

38.4.1.8.2 fobi.urls.edit module

38.4.1.8.3 fobi.urls.view module

38.4.1.8.4 Module contents

38.4.1.9 fobi.wizard package

38.4.1.9.1 Subpackages

38.4.1.9.1.1 fobi.wizard.views package

38.4.1.9.1.2 Submodules

38.4.1.9.1.3 fobi.wizard.views.dynamic module

```

class fobi.wizard.views.dynamic.DynamicWizardView(**kwargs)
    Bases: django.views.generic.base.TemplateView

    The WizardView is used to create multi-page forms.

    Handles all the storage and validation stuff. The wizard is based on Django's generic class based views.

```

classmethod as_view (*args, **kwargs)

As view.

This method is used within `urls.py` to create unique `wizardview` instances for every request. We need to override this method because we add some kwargs which are needed to make the `wizardview` usable.

compute_form_list (form_list=None, *args, **kwargs)

Compute the forms list.

condition_dict = None

dispatch (request, *args, **kwargs)

Dispatch.

This method gets called by the routing engine. The first argument is *request* which contains a *HttpRequest* instance. The request is stored in *self.request* for later use. The storage instance is stored in *self.storage*.

After processing the request using the *dispatch* method, the response gets updated by the storage engine (for example add cookies).

done (form_list, **kwargs)

Done.

This method must be overridden by a subclass to process the form data after processing all steps.

get (request, *args, **kwargs)

GET requests.

This method handles GET requests.

If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to restart the process. The data of the wizard will be resetted before rendering the first step

get_all_cleaned_data ()

Get all cleaned data.

Returns a merged dictionary of all step `cleaned_data` dictionaries. If a step contains a *FormSet*, the key will be prefixed with 'formset-' and contain a list of the formset `cleaned_data` dictionaries.

get_cleaned_data_for_step (step)

Get clean data for step.

Returns the cleaned data for a given *step*. Before returning the cleaned data, the stored values are revalidated through the form. If the data doesn't validate, `None` will be returned.

get_context_data (form, **kwargs)

Get context data.

Returns the template context for a step. You can overwrite this method to add more data for all or some steps. This method returns a dictionary containing the rendered form step. Available template context variables are:

- all extra data stored in the storage backend
- *wizard* - a dictionary representation of the wizard instance

Example:

```
class MyWizard(WizardView):
    def get_context_data(self, form, **kwargs):
        context = super(MyWizard, self).get_context_data(form=form,
                                                         **kwargs)

        if self.steps.current == 'my_step_name':
```

(continues on next page)

(continued from previous page)

```
context.update({'another_var': True})
return context
```

get_form (*step=None, data=None, files=None*)

Get the form.

Constructs the form for a given *step*. If no *step* is defined, the current step will be determined automatically.

The form will be initialized using the *data* argument to prefill the new form. If needed, instance or queryset (for *ModelForm* or *ModelFormSet*) will be added too.

get_form_initial (*step*)

Get form initial

Returns a dictionary which will be passed to the form for *step* as *initial*. If no initial data was provided while initializing the form wizard, an empty dictionary will be returned.

get_form_instance (*step*)

Get form instance.

Returns an object which will be passed to the form for *step* as *instance*. If no instance object was provided while initializing the form wizard, *None* will be returned.

get_form_kwargs (*step=None*)

Get form kwargs.

Returns the keyword arguments for instantiating the form (or formset) on the given step.

get_form_list ()

Get form list.

This method returns a *form_list* based on the initial form list but checks if there is a condition method/value in the *condition_list*. If an entry exists in the condition list, it will call/read the value and respect the result. (True means add the form, False means ignore the form)

The *form_list* is always generated on the fly because condition methods could use data from other (maybe previous forms).

get_form_prefix (*step=None, form=None*)

Get form prefix.

Returns the prefix which will be used when calling the actual form for the given step. *step* contains the step-name, *form* the form which will be called with the returned prefix.

If no step is given, the *form_prefix* will determine the current step automatically.

get_form_step_data (*form*)

Get form step data.

Is used to return the raw form data. You may use this method to manipulate the data.

get_form_step_files (*form*)

Get form step files.

Is used to return the raw form files. You may use this method to manipulate the data.

get_initial_wizard_data (**args, **kwargs*)

This should be implemented in your subclass.

You are supposed to return a dict with the dynamic properties, such as *form_list* or *template_name*.

classmethod get_initkwargs (*form_list=None, initial_dict=None, instance_dict=None, condition_dict=None, *args, **kwargs*)

Create a dict with all needed parameters.

For the form wizard instances.

- *form_list* - is a list of forms. The list entries can be single form classes or tuples of (*step_name, form_class*). If you pass a list of forms, the wizardview will convert the class list to (*zero_based_counter, form_class*). This is needed to access the form for a specific step.
- *initial_dict* - contains a dictionary of initial data dictionaries. The key should be equal to the *step_name* in the *form_list* (or the str of the zero based counter - if no *step_names* added in the *form_list*)
- *instance_dict* - contains a dictionary whose values are model instances if the step is based on a `ModelForm` and querysets if the step is based on a `ModelFormSet`. The key should be equal to the *step_name* in the *form_list*. Same rules as for *initial_dict* apply.
- *condition_dict* - contains a dictionary of boolean values or callables. If the value of for a specific *step_name* is callable it will be called with the wizardview instance as the only argument. If the return value is true, the step's form will be used.

get_next_step (*step=None*)

Get next step.

Returns the next step after the given *step*. If no more steps are available, `None` will be returned. If the *step* argument is `None`, the current step will be determined automatically.

get_prefix (*request, *args, **kwargs*)

Get prefix.

get_prev_step (*step=None*)

Get previous step.

Returns the previous step before the given *step*. If there are no steps available, `None` will be returned. If the *step* argument is `None`, the current step will be determined automatically.

get_step_index (*step=None*)

Get step index.

Returns the index for the given *step* name. If no step is given, the current step will be used to get the index.

initial_dict = None

instance_dict = None

post (**args, **kwargs*)

POST requests.

This method handles POST requests.

The wizard will render either the current step (if form validation wasn't successful), the next step (if the current step was stored successful) or the done view (if no more steps are available)

process_step (*form*)

Process the step.

This method is used to post-process the form data. By default, it returns the raw *form.data* dictionary.

process_step_files (*form*)

Process step files.

This method is used to post-process the form files. By default, it returns the raw *form.files* dictionary.

render (*form=None*, ***kwargs*)

Render.

Returns a `HttpResponse` containing all needed context data.

render_done (*form*, ***kwargs*)

Render done.

This method gets called when all forms passed. The method should also re-validate all steps to prevent manipulation. If any form fails to validate, *render_revalidation_failure* should get called. If everything is fine call *done*.

render_goto_step (*goto_step*, ***kwargs*)

Render goto step.

This method gets called when the current step has to be changed. *goto_step* contains the requested step to go to.

render_next_step (*form*, ***kwargs*)

Render next step.

This method gets called when the next step/form should be rendered. *form* contains the last/current form.

render_revalidation_failure (*step*, *form*, ***kwargs*)

Render revalidation failure.

Gets called when a form doesn't validate when rendering the done view. By default, it changes the current step to failing forms step and renders the form.

storage_name = `None`

template_name = `'formtools/wizard/wizard_form.html'`

class `fobi.wizard.views.dynamic.DynamicSessionWizardView` (***kwargs*)

Bases: `fobi.wizard.views.dynamic.DynamicWizardView`

A WizardView with pre-configured SessionStorage backend.

storage_name = `'formtools.wizard.storage.session.SessionStorage'`

class `fobi.wizard.views.dynamic.DynamicCookieWizardView` (***kwargs*)

Bases: `fobi.wizard.views.dynamic.DynamicWizardView`

A WizardView with pre-configured CookieStorage backend.

storage_name = `'formtools.wizard.storage.cookie.CookieStorage'`

class `fobi.wizard.views.dynamic.DynamicNamedUrlWizardView` (***kwargs*)

Bases: `fobi.wizard.views.dynamic.DynamicWizardView`

A WizardView with URL named steps support.

done_step_name = `None`

get (**args*, ***kwargs*)

GET request.

This renders the form or, if needed, does the http redirects.

get_context_data (*form*, ***kwargs*)

Get context data.

`NamedUrlWizardView` provides the `url_name` of this wizard in the context dict *wizard*.

classmethod `get_initkwargs (*args, **kwargs)`

Get init kwargs.

We require a `url_name` to reverse URLs later. Additionally users can pass a `done_step_name` to change the URL name of the “done” view.

get_step_url (*step*)

Get step URL.

post (**args, **kwargs*)

POST request.

Do a redirect if user presses the prev. step button. The rest of this is super’d from WizardView.

render_done (*form, **kwargs*)

Render done.

When rendering the done view, we have to redirect first (if the URL name doesn’t fit).

render_goto_step (*goto_step, **kwargs*)

Render goto step.

This method gets called when the current step has to be changed. *goto_step* contains the requested step to go to.

render_next_step (*form, **kwargs*)

Render next step.

When using the NamedUrlWizardView, we have to redirect to update the browser’s URL to match the shown step.

render_revalidation_failure (*failed_step, form, **kwargs*)

Render revalidation failure.

When a step fails, we have to redirect the user to the first failing step.

url_name = None

class `fobi.wizard.views.dynamic.DynamicNamedUrlSessionWizardView (**kwargs)`

Bases: `fobi.wizard.views.dynamic.DynamicNamedUrlWizardView`

A NamedUrlWizardView with pre-configured SessionStorage backend.

storage_name = 'formtools.wizard.storage.session.SessionStorage'

class `fobi.wizard.views.dynamic.DynamicNamedUrlCookieWizardView (**kwargs)`

Bases: `fobi.wizard.views.dynamic.DynamicNamedUrlWizardView`

A NamedUrlFormWizard with pre-configured CookieStorageBackend.

storage_name = 'formtools.wizard.storage.cookie.CookieStorage'

38.4.1.9.1.4 fobi.wizard.views.views module

class `fobi.wizard.views.views.WizardView (**kwargs)`

Bases: `formtools.wizard.views.WizardView`, `fobi.wizard.views.views.PatchGetMixin`

Patched version of the original WizardView.

get (*request, *args, **kwargs*)

GET requests.

This method handles GET requests.

If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to restart the process. The data of the wizard will be resetted before rendering the first step.

```
class fobi.wizard.views.views.SessionWizardView (**kwargs)
    Bases: formtools.wizard.views.SessionWizardView, fobi.wizard.views.views.
    PatchGetMixin
```

A WizardView with pre-configured SessionStorage backend.

```
get (request, *args, **kwargs)
    GET requests.
```

This method handles GET requests.

If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to restart the process. The data of the wizard will be resetted before rendering the first step.

```
class fobi.wizard.views.views.CookieWizardView (**kwargs)
    Bases: formtools.wizard.views.CookieWizardView, fobi.wizard.views.views.
    PatchGetMixin
```

A WizardView with pre-configured CookieStorage backend.

```
get (request, *args, **kwargs)
    GET requests.
```

This method handles GET requests.

If a GET request reaches this point, the wizard assumes that the user just starts at the first step or wants to restart the process. The data of the wizard will be resetted before rendering the first step.

38.4.1.9.1.5 Module contents

38.4.1.9.2 Module contents

38.4.2 Submodules

38.4.3 fobi.admin module

```
fobi.admin.base_bulk_change_plugins (PluginForm, named_url, modeladmin, request, queryset)
    Bulk change of plugins action additional view.
```

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

Base plugin admin.

```
class Meta
```

Bases: object

Meta.

```
app_label = <django.utils.functional.lazy.<locals>.__proxy__ object>
```

```
bulk_change_plugins (request)
    Bulk change plugins.
```

This is where the data is actually processed.

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

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

```

    get_queryset (request)
        Tweak queryset.

    has_add_permission (request)
        Has add permissions.

        We don't want to allow to add form elements/handlers manually. It should happen using the management
        command fobi_sync_plugins instead.

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

    media

    readonly_fields = ('plugin_uid', 'plugin_uid_admin')

fobi.admin.bulk_change_form_element_plugins (modeladmin, request, queryset)
    Bulk change FormElement plugins.

fobi.admin.bulk_change_form_handler_plugins (modeladmin, request, queryset)
    Bulk change FormHandler plugins.

fobi.admin.bulk_change_form_wizard_handler_plugins (modeladmin, request, queryset)
    Bulk change FormWizardHandler plugins.

class fobi.admin.FormElementAdmin (model, admin_site)
    Bases: fobi.admin.BasePluginModelAdmin

    FormElement admin.

    actions = [<function bulk_change_form_element_plugins>]

    get_urls ()
        Get URLs.

    media

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

    FormElementEntry admin.

    class Meta
        Bases: object

        Meta.

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

    fieldsets = ((<django.utils.functional.lazy.<locals>.__proxy__ object>, {'fields': ('
    get_queryset (request)
        Get queryset.

    list_display = ('plugin_uid', 'plugin_uid_code', 'plugin_data', 'position', 'form_entry
    list_editable = ('position',)
    list_filter = ('form_entry', 'plugin_uid')

    media

    readonly_fields = ('plugin_uid_code',)

class fobi.admin.FormElementEntryInlineAdmin (parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    FormElementEntry inline admin.

```

```

    extra = 0

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

    form
        alias of fobi.forms.FormElementEntryForm

    media

    model
        alias of fobi.models.FormElementEntry

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

    FormEntry admin.

    class Meta
        Bases: object

        Meta.

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

    fieldsets = ((<django.utils.functional.lazy.<locals>.__proxy__ object>, {'fields': ('
    inlines = [<class 'fobi.admin.FormElementEntryInlineAdmin'>, <class 'fobi.admin.FormHan
    list_display = ('name', 'slug', 'user', 'is_public', 'is_active', 'created', 'updated'
    list_editable = ('is_public', 'is_cloneable')
    list_filter = ('is_public', 'is_cloneable')

    media

    radio_fields = {'user': 2}

    readonly_fields = ('slug',)

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

    FormEieldsetEntry admin.

    class Meta
        Bases: object

        Meta.

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

    fieldsets = ((None, {'fields': ('form_entry', 'name', 'is_repeatable')})),)

    list_display = ('form_entry', 'name', 'is_repeatable')

    list_editable = ('is_repeatable',)

    list_filter = ('is_repeatable',)

    media

class fobi.admin.FormHandlerAdmin(model, admin_site)
    Bases: fobi.admin.BasePluginModelAdmin

    FormHandler admin.

    actions = [<function bulk_change_form_handler_plugins>]

```

```
    get_urls()
        Get URLs.

    media

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

    FormHandlerEntry admin.

    class Meta
        Bases: object

        Meta.

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

    fieldsets = ((<django.utils.functional.lazy.<locals>.__proxy__ object>, {'fields': ('
get_queryset(request)
    Tweak the queryset.

    list_display = ('plugin_uid', 'plugin_uid_code', 'plugin_data', 'form_entry')

    list_filter = ('form_entry', 'plugin_uid')

    media

    readonly_fields = ('plugin_uid_code',)

class fobi.admin.FormHandlerEntryInlineAdmin(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline

    FormHandlerEntry inline admin.

    extra = 0

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

    form
        alias of fobi.forms.FormHandlerEntryForm

    media

    model
        alias of fobi.models.FormHandlerEntry

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

    FormWizardEntry admin.

    class Meta
        Bases: object

        Meta.

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

    fieldsets = ((<django.utils.functional.lazy.<locals>.__proxy__ object>, {'fields': ('
    inlines = [<class 'fobi.admin.FormWizardFormEntryInlineAdmin'>, <class 'fobi.admin.For
    list_display = ('name', 'slug', 'user', 'is_public', 'created', 'updated', 'is_cloneab
    list_editable = ('is_public', 'is_cloneable')

    list_filter = ('is_public', 'is_cloneable')
```

```

media
radio_fields = {'user': 2}
readonly_fields = ('slug',)

class fobi.admin.FormWizardFormEntryInlineAdmin(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
    FormWizardFormEntry inline admin.

    extra = 0
    fields = ('form_entry', 'position')
    media
    model
        alias of fobi.models.FormWizardFormEntry

class fobi.admin.FormWizardHandlerAdmin(model, admin_site)
    Bases: fobi.admin.BasePluginModelAdmin
    FormHandler admin.

    actions = [<function bulk_change_form_wizard_handler_plugins>]
    get_urls()
        Get URLs.
    media

class fobi.admin.FormWizardHandlerEntryInlineAdmin(parent_model, admin_site)
    Bases: django.contrib.admin.options.TabularInline
    FormWizardHandlerEntry inline admin.

    extra = 0
    fields = ('plugin_uid', 'plugin_data')
    form
        alias of fobi.forms.FormWizardHandlerEntryForm
    media
    model
        alias of fobi.models.FormWizardHandlerEntry

```

38.4.4 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*

38.4.5 fobi.apps module

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

38.4.6 fobi.base module

Base module. All *uids* are supposed to be pythonic function names (see PEP <http://www.python.org/dev/peps/pep-0008/#function-names>).

```
fobi.base.assemble_form_field_widget_class(base_class, plugin)
    Assemble form field widget class.
    Finish this or remove.
    #TODO
```

```
class fobi.base.BaseDataStorage
    Bases: object
    Base storage data.
```

```
class fobi.base.BaseFormFieldPluginForm
    Bases: fobi.base.BasePluginForm
    Base form for form field plugins.
    help_text = <django.forms.fields.CharField object>
    label = <django.forms.fields.CharField object>
    name = <django.forms.fields.CharField object>
    plugin_data_fields = [('name', ''), ('label', ''), ('help_text', ''), ('required', False)]
    required = <django.forms.fields.BooleanField object>
    validate_plugin_data(form_element_entries, request=None)
        Validate plugin data.
```

Parameters

- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry*.
- **request** (*django.http.HttpRequest*) –

Return *bool*

class fobi.base.BasePlugin (user=None)

Bases: object

Base plugin.

Base form field from which every form field should inherit.

Properties

- **uid (string): Plugin uid (obligatory).** Example value: ‘dummy’, ‘wysiwyg’, ‘news’.
- **name (string): Plugin name (obligatory).** Example value: ‘Dummy plugin’, ‘WYSIWYG’, ‘Latest news’.
- **description (string): Plugin decription (optional).** Example value: ‘Dummy plugin used just for testing’.
- **help_text (string): Plugin help text (optional).** This text would be shown in fobi.views.add_form_plugin_entry and fobi.views.edit_form_plugin_entry views.
- **form: Plugin form (optional).** A subclass of `django.forms.Form`. Should be given in case plugin is configurable.
- **add_form_template (str) (optional): Add form template (optional).** If given, overrides the `fobi.views.add_form_handler_entry` default template.
- **edit_form_template (string): Edit form template (optional).** If given, overrides the `fobi.views.edit_form_handler_entry` default template.
- **html_classes (list):** List of extra HTML classes for the plugin.
- **group (string): Plugin are grouped under the specified group.** Override in your plugin if necessary.

add_form_template = None

clone_plugin_data (entry)

Clone plugin data.

Used when copying entries. If any objects or files are created by plugin, they should be cloned.

Parameters `fobi.models.AbstractPluginEntry` – Instance of `fobi.models.AbstractPluginEntry`.

Return string JSON dumped string of the cloned plugin data. The returned value would be inserted as is into the `fobi.models.AbstractPluginEntry.plugin_data` field.

delete_plugin_data ()

Delete plugin data (internal method).

Used in `fobi.views.delete_form_entry` and `fobi.views.delete_form_handler_entry`. Fired automatically, when `fobi.models.FormEntry` object is about to be deleted. Make use of it if your plugin creates database records or files that are not monitored externally but by fobi only.

description = None

edit_form_template = None

form = None

get_cloned_plugin_data (*update={}*)

Get cloned plugin data.

Get the cloned plugin data and returns it in a JSON dumped format.

Parameters *update* (*dict*) –

Return string JSON dumped string of the cloned plugin data.

Example

In the **get_cloned_plugin_data** method of your plugin, do as follows:

```
>>> def clone_plugin_data(self, dashboard_entry):
>>>     cloned_image = clone_file(self.data.image, relative_path=True)
>>>     return self.get_cloned_plugin_data(
>>>         update={'image': cloned_image}
>>>     )
```

get_form ()

Get the plugin form class.

Override this method in your subclassed `fobi.base.BasePlugin` class when you need your plugin setup to vary depending on the placeholder, workspace, user or request given. By default returns the value of the `form` attribute defined in your plugin.

Return `django.forms.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.lazy.<locals>.__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, redefine this method in your plugin. If you need your data to be periodically updated, add a cron-job which would run `fobi_update_plugin_data` management command (see `fobi.management.commands.fobi_update_plugin_data` module).

Parameters or **fobi.models.FormHandlerEntry** (*fobi.models.FormElementEntry*) – Instance of `fobi.models.FormHandlerEntry`.

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

widget = None

class `fobi.base.BasePluginForm`

Bases: `object`

Not a form actually; defined for magic only.

Property iterable plugin_data_fields Fields to get when calling the `get_plugin_data` method. These field will be JSON serialized. All other fields, even if they are part of the form, won't be. Make sure all fields are serializable. If some of them aren't, override the `save_plugin_data` method and make them serializable there. See `fobi.contrib.plugins.form_elements.fields.select.forms` as a good example.

Example

```
>>> plugin_data_fields = (
>>>     ('name', ''),
>>>     ('active': False)
>>> )
```

get_plugin_data (*request=None, json_format=True*)

Get plugin data.

Data that would be saved in the `plugin_data` field of the `fobi.models.FormElementEntry` or `“fobi.models.FormHandlerEntry“` subclassed model.

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

plugin_data_fields = None

save_plugin_data (*request=None*)

Save plugin data.

Dummy, but necessary.

validate_plugin_data (*form_element_entries, request=None*)

Validate plugin data.

Parameters

- **form_element_entries** (*iterable*) – Iterable of `fobi.models.FormElementEntry`.
- **request** (*django.http.HttpRequest*) –

Return bool

class fobi.base.BaseRegistry

Bases: object

Base registry.

Registry of plugins. It's essential, that class registered has the `uid` property.

If `fail_on_missing_plugin` is set to `True`, an appropriate exception (`plugin_not_found_exception_cls`) is raised in cases if plugin could't be found in the registry.

Property mixed type

Property bool fail_on_missing_plugin

Property fobi.exceptions.DoesNotExist plugin_not_found_exception_cls

Property str plugin_not_found_error_message

fail_on_missing_plugin = False

get (*uid, default=None*)

Get the given entry from the registry.

Parameters

- **uid** (*string*) –

- **default** (*mixed*) –
:return mixed.

items ()
Shortcut to self._registry.items().

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

plugin_not_found_exception_cls
alias of *fobi.exceptions.DoesNotExist*

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

Parameters

- **cls** (*mixed*) –
- **force** (*bool*) –

registry
Shortcut to self._registry.

type = None

unregister (*cls*)
Un-register.

class *fobi.base.ClassProperty*
Bases: *property*
ClassProperty.

fobi.base.classproperty
alias of *fobi.base.ClassProperty*

fobi.base.collect_plugin_media (*form_element_entries*, *request=None*)
Collect the plugin media for form element entries given.

Parameters

- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* instances.
- **request** (*django.http.HttpRequest*) –

Return dict Returns a dict containing the ‘js’ and ‘css’ keys. Correspondent values of those keys are lists containing paths to the CSS and JS media files.

fobi.base.ensure_autodiscover ()
Ensure that plugins are auto-discovered.

The form callbacks registry is intentionally left out, since they will be auto-discovered in any case if other modules are discovered.

fobi.base.fire_form_callbacks (*form_entry*, *request*, *form*, *stage=None*)
Fire form callbacks.

Parameters

- **form_entry** (*fobi.models.FormEntry*) –
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –

- **stage** (*string*) –

Return `django.forms.Form` form

class `fobi.base.FormCallback`

Bases: `fobi.base.BaseFormCallback`

Form callback.

callback (*form_entry, request, form*)

Callback.

Custom callback code should be implemented here.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

class `fobi.base.FormCallbackRegistry`

Bases: `object`

Registry of callbacks.

Holds callbacks for stages listed in the `fobi.constants.CALLBACK_STAGES`.

get_callbacks (*stage=None*)

Get callbacks for the stage given.

Parameters **stage** (*string*) –

Return `list`

register (*cls*)

Registers the plugin in the registry.

Parameters **cls** (*mixed*) –

uidfy (*cls*)

Makes a UID string from the class given.

Parameters **cls** (*mixed*) –

Return `string`

class `fobi.base.FormElementPlugin` (*user=None*)

Bases: `fobi.base.BasePlugin`

Base form element plugin.

Property `fobi.base.FormElementPluginDataStorage` **storage**

Property `bool` **has_value** If set to `False`, ignored (removed) from the POST when processing the form.

get_custom_field_instances (*integrate_with, request=None, form_entry=None, form_element_entries=None, has_value=None, **kwargs*)

Get custom field instances.

Parameters

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

- **form_entry** –
- **form_element_entries** –
- **has_value** (*bool*) – If not None, used for filtering out.

Return list

get_form_field_instances (*request=None, form_entry=None, form_element_entries=None, **kwargs*)

Get the instances of form fields, that plugin contains.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_entry** (*fobi.models.FormEntry*) –
- **form_element_entries** (*django.db.models.QuerySet*) – Queryset of *fobi.models.FormElementEntry* instances.

Return list List of Django form field instances.

Example

```
>>> from django.forms.fields import CharField, IntegerField, TextField
>>> [CharField(max_length=100), IntegerField(), TextField()]
```

get_origin_kwargs_update_func_results (*kwargs_update_func, form_element_entry, origin, extra={}, widget_cls=None*)

Get origin kwargs update func results.

If *kwargs_update_func* is given, is callable and returns results without failures, return the result. Otherwise - return None.

get_origin_return_func_results (*return_func, form_element_entry, origin*)

Get origin return func results.

If *return_func* is given, is callable and returns results without failures, return the result. Otherwise - return None.

has_value = False

is_hidden = False

storage

alias of *FormElementPluginDataStorage*

submit_plugin_form_data (*form_entry, request, form, form_element_entries=None, **kwargs*)

Submit plugin form data.

Called on form submission (when user actually posts the data to assembled form).

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –
- **form_element_entries** (*iterable*) –

class *fobi.base.FormElementPluginDataStorage*

Bases: *fobi.base.BaseDataStorage*

Storage for *FormElementPlugin* data.

class fobi.base.**FormElementPluginRegistry**

Bases: *fobi.base.BaseRegistry*

Form element plugins registry.

fail_on_missing_plugin = True

plugin_not_found_exception_cls

alias of *fobi.exceptions.FormElementPluginDoesNotExist*

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

class fobi.base.**FormElementPluginWidget** (*plugin*)

Bases: *fobi.base.BasePluginWidget*

Form element plugin widget.

storage

alias of *FormElementPluginWidgetDataStorage*

class fobi.base.**FormElementPluginWidgetRegistry**

Bases: *fobi.base.BasePluginWidgetRegistry*

Registry of form element plugins.

type

alias of *FormElementPluginWidget*

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

Bases: *fobi.base.FormElementPlugin*

Form field plugin.

has_value = True

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

Bases: *fobi.base.BasePlugin*

Form handler plugin.

Property *fobi.base.FormHandlerPluginDataStorage* **storage**

Property **bool allow_multiple** If set to True, plugin can be used multiple times within (per form). Otherwise - just once.

allow_multiple = True

custom_actions (*form_entry, request=None*)

Custom actions.

Override this method in your form handler if you want to specify custom actions. Note, that expected return value of this method is an iterable with a triple, where the first item is the URL of the action and the second item is the action title and the third item is the icon class of the action.

Example

```
>>> return (
>>>     ('/add-to-favorites/',
>>>      'Add to favourites',
>>>      'glyphicon glyphicon-favourties'),
>>> )
```

get_custom_actions (*form_entry, request=None*)

Internal method to for obtaining the *get_custom_actions*.

run (*form_entry*, *request*, *form*, *form_element_entries=None*)

Run.

Custom code should be implemented here.

Parameters

- **form_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

Return mixed May be a tuple (bool, mixed) or None

run_integration_handler (*integrate_with*, *form_entry*, *request*, *form_element_entries=None*, ***kwargs*)

Run integration handler.

storage

alias of *FormHandlerPluginDataStorage*

class *fobi.base.FormHandlerPluginDataStorage*

Bases: *fobi.base.BaseDataStorage*

Storage for *FormHandlerPlugin* data.

class *fobi.base.FormHandlerPluginRegistry*

Bases: *fobi.base.BaseRegistry*

Form handler plugins registry.

fail_on_missing_plugin = True

plugin_not_found_exception_cls

alias of *fobi.exceptions.FormHandlerPluginDoesNotExist*

type

alias of *FormHandlerPlugin*

class *fobi.base.FormHandlerPluginWidget* (*plugin*)

Bases: *fobi.base.BasePluginWidget*

Form handler plugin widget.

storage

alias of *FormHandlerPluginWidgetDataStorage*

class *fobi.base.FormHandlerPluginWidgetRegistry*

Bases: *fobi.base.BasePluginWidgetRegistry*

Registry of form handler plugins.

type

alias of *FormHandlerPluginWidget*

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

Bases: *fobi.base.BasePlugin*

Form wizard handler plugin.

Property *fobi.base.FormWizardHandlerPluginDataStorage* storage

Property bool allow_multiple If set to True, plugin can be used multiple times within (per form). Otherwise - just once.

DONE

allow_multiple = True

custom_actions (*form_wizard_entry, request=None*)

Custom actions.

Override this method in your form handler if you want to specify custom actions. Note, that expected return value of this method is an iterable with a triple, where the first item is the URL of the action and the second item is the action title and the third item is the icon class of the action.

Example

```
>>> return (
>>>     ('/add-to-favorites/',
>>>     'Add to favourites',
>>>     'glyphicon glyphicon-favourties'),
>>> )
```

get_custom_actions (*form_wizard_entry, request=None*)

Internal method to for obtaining the get_custom_actions.

run (*form_wizard_entry, request, form_list, form_wizard, form_element_entries=None*)

Run.

Custom code should be implemented here.

Parameters

- **form_wizard_entry** (*fobi.models.FormEntry*) – Instance of *fobi.models.FormEntry*.
- **request** (*django.http.HttpRequest*) –
- **form** (*django.forms.Form*) –
- **form_element_entries** (*iterable*) – Iterable of *fobi.models.FormElementEntry* objects.

Return mixed May be a tuple (bool, mixed) or None

storage

alias of *FormWizardHandlerPluginDataStorage*

class *fobi.base.FormWizardHandlerPluginDataStorage*

Bases: *fobi.base.BaseDataStorage*

Storage for *FormWizardHandlerPlugin* handler data.

class *fobi.base.FormWizardHandlerPluginRegistry*

Bases: *fobi.base.BaseRegistry*

Form wizard handler plugins registry.

fail_on_missing_plugin = True

plugin_not_found_exception_cls

alias of *fobi.exceptions.FormWizardHandlerPluginDoesNotExist*

type

alias of *FormWizardHandlerPlugin*

class fobi.base.**FormWizardHandlerPluginWidget** (*plugin*)

Bases: fobi.base.BasePluginWidget

Form wizard handler plugin widget.

storage

alias of FormWizardHandlerPluginWidgetDataStorage

class fobi.base.**FormWizardHandlerPluginWidgetRegistry**

Bases: fobi.base.BasePluginWidgetRegistry

Registry of form wizard handler plugins.

type

alias of *FormWizardHandlerPluginWidget*

fobi.base.**get_form_element_plugin_widget** (*plugin_uid*, *request=None*, *as_instance=False*,
theme=None)

Get the form element plugin widget for the *plugin_uid* given.

Parameters

- **plugin_uid** (*str*) – UID of the plugin to get the widget for.
- **request** (*django.http.HttpRequest*) –
- **as_instance** (*bool*) –
- **theme** (*fobi.base.BaseTheme*) – Subclass of.

Return BasePluginWidget Subclass of.

fobi.base.**get_form_handler_plugin_widget** (*plugin_uid*, *request=None*, *as_instance=False*,
theme=None)

Get the form handler plugin widget for the *plugin_uid* given.

Parameters

- **plugin_uid** (*str*) – UID of the plugin to get the widget for.
- **request** (*django.http.HttpRequest*) –
- **as_instance** (*bool*) –
- **theme** (*fobi.base.BaseTheme*) – Subclass of.

Return BasePluginWidget Subclass of.

fobi.base.**get_form_wizard_handler_plugin_widget** (*plugin_uid*, *request=None*,
as_instance=False, *theme=None*)

Get the form wizard handler plugin widget for the *plugin_uid* given.

Parameters

- **plugin_uid** (*str*) – UID of the plugin to get the widget for.
- **request** (*django.http.HttpRequest*) –
- **as_instance** (*bool*) –
- **theme** (*fobi.base.BaseTheme*) – Subclass of.

Return BasePluginWidget Subclass of.

fobi.base.**get_ordered_form_handlers** ()

Get ordered form handler plugins.

Gets form handler plugins in the execution order as a sortable dictionary, which can be later on used to add real plugins to be executed.

Return `fobi.data_structures.SortableDict`

`fobi.base.get_ordered_form_wizard_handlers()`

Get ordered form wizard_handler plugins.

Gets form wizard handler plugins in the execution order as a sortable dictionary, which can be later on used to add real plugins to be executed.

Return `fobi.data_structures.SortableDict`

`fobi.base.get_plugin_widget(registry, plugin_uid, request=None, as_instance=False, theme=None)`

Get the plugin widget for the `plugin_uid` given.

Looks up in the `registry` provided.

Parameters

- **registry** (`fobi.base.BasePluginWidgetRegistry`) – Subclass of.
- **plugin_uid** (`str`) – UID of the plugin to get the widget for.
- **request** (`django.http.HttpRequest`) –
- **as_instance** (`bool`) –
- **theme** (`fobi.base.BaseTheme`) – Subclass of.

Return `BasePluginWidget` Subclass of.

`fobi.base.get_processed_form_data(form, form_element_entries)`

Gets processed form data.

Simply fires both `fobi.base.get_cleaned_data` and `fobi.base.get_field_name_to_label_map` functions and returns the result.

Parameters

- **form** (`django.forms.Form`) –
- **form_element_entries** (`iterable`) – Iterable of form element entries.

Return tuple

`fobi.base.get_processed_form_wizard_data(form_wizard, form_list, form_element_entries)`

Get processed form wizard data.

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

Get registered form callbacks for the stage given.

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

Get registered form element plugin uids.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_element_plugins()`

Get registered form element plugins.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_element_plugins_grouped(sort_by_value=False)`

Get registered form element plugins grouped.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return dict

`fobi.base.get_registered_form_handler_plugin_uids(flattern=True)`

Get registered form handler plugin uids.

Gets a list of UIDs of registered form handler plugins. If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_handler_plugins(as_instances=False)`

Get registered form handler plugins.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_wizard_handler_plugin_uids(flattern=True)`

Get registered form handler plugin uids.

Gets a list of UIDs of registered form wizard handler plugins. If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_form_wizard_handler_plugins(as_instances=False)`

Get registered form handler wizard plugins.

Gets a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_element_plugin_uids(flattern=True)`

Get registered custom field instance plugin uids.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_element_plugins()`

Get registered custom field instance plugins.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_registered_integration_form_element_plugins_grouped()`

Get registered custom field instance plugins grouped.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return dict

`fobi.base.get_registered_integration_form_handler_plugin_uids(flattern=True)`

Get registered integration form handler plugin uids.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

```
fobi.base.get_registered_integration_form_handler_plugins()
```

Get registered integration form handler plugins.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

```
fobi.base.get_registered_integration_form_handler_plugins_grouped()
```

Get registered integration form handler plugins grouped.

Gets a list of registered plugins in a form if tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return dict

```
fobi.base.get_registered_plugin_uids(registry, flatten=True, sort_items=True)
```

Get a list of registered plugin uids as a list .

If not yet auto-discovered, auto-discovers them.

The *sort_items* is applied only if *flatten* is True.

Parameters

- **registry** –
- **flatten** (*bool*) –
- **sort_items** (*bool*) –

Return list

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

Get registered plugins.

Get a list of registered plugins in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Parameters

- **registry** –
- **as_instances** (*bool*) –
- **sort_items** (*bool*) –

Return list

```
fobi.base.get_registered_theme_uids(flatten=True)
```

Get registered theme uids.

Gets a list of registered themes in a form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

```
fobi.base.get_registered_themes()
```

Get registered themes.

Gets a list of registered themes in form of tuple (plugin name, plugin description). If not yet auto-discovered, auto-discovers them.

Return list

`fobi.base.get_theme(request=None, theme_uid=None, as_instance=False)`

Get theme.

Gets the theme by `theme_uid` given. If left empty, takes the default one chosen in settings module.

Raises a `fobi.exceptions.ThemeDoesNotExist` when no default layout could be found.

Parameters

- **request** (`django.http.HttpRequest`) –
- **theme_uid** (`int`) –
- **as_instance** (`bool`) –

Return `fobi.base.BaseTheme` Subclass of `fobi.base.BaseTheme`.

class `fobi.base.IntegrationFormCallback`

Bases: `object`

Integration form callback.

callback (`form_entry, request, **kwargs`)

Callback.

Custom callback code should be implemented here.

Parameters

- **form_entry** (`fobi.models.FormEntry`) – Instance of `fobi.models.FormEntry`.
- **request** (`django.http.HttpRequest`) –
- **form** (`django.forms.Form`) –

integrate_with = `None`

class `fobi.base.IntegrationFormCallbackRegistry`

Bases: `object`

Registry of callbacks for integration plugins.

Holds callbacks for stages listed in the `fobi.constants.CALLBACK_STAGES`.

get_callbacks (`integrate_with, stage=None`)

Get callbacks for the stage given.

Parameters

- **integrate_with** (`str`) –
- **stage** (`string`) –

Return list

register (`cls`)

Registers the plugin in the registry.

Parameters `cls` (`mixed`) –

registry

uidfy (`cls`)

Makes a UID string from the class given.

Parameters `cls` (`mixed`) –

Return string

```
class fobi.base.IntegrationFormElementPlugin (user=None)
    Bases: fobi.base.BasePlugin

    Base custom field instance plugin for integration.

    has_value = False

    is_hidden = False

    storage
        alias of IntegrationFormElementPluginDataStorage

class fobi.base.IntegrationFormElementPluginDataStorage
    Bases: fobi.base.BaseDataStorage

    Storage for IntegrationFormElementPlugin.

class fobi.base.IntegrationFormElementPluginProcessor (*args, **kwargs)
    Bases: object

    Custom form field instance processor.

    Supposed to have implemented a single method called process_custom_form_field_instance.

    process_custom_form_field_instance (form_element_entry, form_entry, request,
                                         form_element_plugin)
        You should implement this method in your implementation.

class fobi.base.IntegrationFormElementPluginRegistry
    Bases: fobi.base.BaseIntegrationPluginRegistry

    Integration form element plugin registry.

    fail_on_missing_plugin = True

    plugin_not_found_exception_cls
        alias of fobi.exceptions.IntegrationFormElementPluginDoesNotExist

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

class fobi.base.IntegrationFormFieldPlugin (user=None)
    Bases: fobi.base.IntegrationFormElementPlugin

    Integration form field plugin for custom field instances.

    has_value = True

class fobi.base.IntegrationFormHandlerPlugin (user=None)
    Bases: fobi.base.BasePlugin

    Base integration form handler plugin for integration.

    has_value = True

    is_hidden = False

    storage
        alias of IntegrationFormHandlerPluginDataStorage

class fobi.base.IntegrationFormHandlerPluginDataStorage
    Bases: fobi.base.BaseDataStorage

    Storage for IntegrationFormHandlerPlugin.
```

class fobi.base.IntegrationFormHandlerPluginRegistry

Bases: fobi.base.BaseIntegrationPluginRegistry

Integration form handler plugin registry.

fail_on_missing_plugin = True

plugin_not_found_exception_cls

alias of `fobi.exceptions.IntegrationFormHandlerPluginDoesNotExist`

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

fobi.base.run_form_handlers(*form_entry*, *request*, *form*, *form_element_entries*=None)

Run form handlers.

Parameters

- **form_entry** (fobi.models.FormEntry) –
- **request** (django.http.HttpRequest) –
- **form** (django.forms.Form) –
- **form_element_entries** (iterable) –

Return tuple List of success responses, list of error responses

fobi.base.run_form_wizard_handlers(*form_wizard_entry*, *request*, *form_list*, *form_wizard*,
form_element_entries=None)

Run form wizard handlers.

Parameters

- **form_wizard_entry** (fobi.models.FormWizardEntry) –
- **request** (django.http.HttpRequest) –
- **form_list** (list) – List of django.forms.Form objects.
- **form_wizard** (fobi.wizard.views.dynamic.DynamicWizardView) – The form wizard view object.
- **form_element_entries** (iterable) – Iterable of fobi.base.FormElementEntry objects.

Return tuple List of success responses, list of error responses

fobi.base.submit_plugin_form_data(*form_entry*, *request*, *form*, *form_element_entries*=None,
**kwargs)

Submit plugin form data for all plugins.

Parameters

- **form_entry** (fobi.models.FormEntry) – Instance of fobi.models.FormEntry.
- **request** (django.http.HttpRequest) –
- **form** (django.forms.Form) –
- **form_element_entries** (iterable) –

fobi.base.validate_form_element_plugin_uid(*plugin_uid*)

Validate the form element plugin uid.

Parameters **plugin_uid** (string) –

Return bool

`fobi.base.validate_form_handler_plugin_uid(plugin_uid)`

Validate the plugin uid.

Parameters `plugin_uid(string)` –

Return bool

`fobi.base.validate_form_wizard_handler_plugin_uid(plugin_uid)`

Validate the plugin uid.

Parameters `plugin_uid(string)` –

Return bool

`fobi.base.validate_integration_form_element_plugin_uid(plugin_uid)`

Validate the custom field instance plugin uid.

Parameters `plugin_uid(string)` –

Return bool

`fobi.base.validate_integration_form_handler_plugin_uid(plugin_uid)`

Validate the integration form handler plugin uid.

Parameters `plugin_uid(string)` –

Return bool

`fobi.base.validate_theme_uid(plugin_uid)`

Validate the theme uid.

Parameters `plugin_uid(string)` –

Return bool

38.4.7 fobi.conf module

`fobi.conf.get_setting(setting, override=None)`

Get setting.

Get a setting from *fobi* conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

Returns Setting value.

38.4.8 fobi.constants module

38.4.9 fobi.context_processors module

`fobi.context_processors.dynamic_values(request)`

Dynamic values exposed to public forms.

`fobi.context_processors.form_importers(request)`

Form importers.

`fobi.context_processors.theme(request)`

Get active theme.

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

Return `fobi.base.BaseTheme` Instance of `fobi.base.BaseTheme`.

38.4.10 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 ()

Iter items (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 (*key*, **args*)

Pop.

popitem ()

Pop item.

setdefault (*key*, *default*)

Set default.

update (*dict_*)

Update.

value_for_index (*index*)

Returns the value of the item at the given zero-based index.

values ()

Iter values (internal method).

38.4.11 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
```

38.4.12 fobi.defaults module

38.4.13 fobi.discover module

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

38.4.14 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,
                                get_form_field_instances_kwargs={})
```

Assemble a form class by given entry.

Parameters

- **form_entry** –
- **base_class** –
- **request** (*django.http.HttpRequest*) –
- **origin** (*string*) –
- **origin_kwargs_update_func** (*callable*) –
- **origin_return_func** (*callable*) –
- **form_element_entries** (*iterable*) – If given, used instead of form_entry. formelemententry_set.all (no additional database hit).
- **get_form_field_instances_kwargs** (*dict*) – To be passed as ****kwargs** to the **method: 'get_form_field_instances_kwargs'**.

```
fobi.dynamic.assemble_form_wizard_class(form_wizard_entry, base_class=<class 'form-
tools.wizard.views.SessionWizardView'>, request=None, origin=None, ori-
origin_kwargs_update_func=None, origin_return_func=None,
form_wizard_form_entries=None, template_name=None)
```

Assemble form wizard class.

Parameters

- **form_wizard_entry** –
- **base_class** –
- **request** –
- **origin** –
- **origin_kwargs_update_func** –
- **origin_return_func** –
- **form_wizard_form_entries** –
- **template_name** –

Returns

38.4.15 fobi.exceptions module

exception fobi.exceptions.BaseException

Bases: Exception

Base exception.

exception `fobi.exceptions.DoesNotExist`
 Bases: `fobi.exceptions.BaseException`
 Raised when something does not exist.

exception `fobi.exceptions.FormCallbackError`
 Bases: `fobi.exceptions.FormPluginError`
 Raised when form callback error occurs.

exception `fobi.exceptions.FormElementPluginDoesNotExist`
 Bases: `fobi.exceptions.PluginDoesNotExist`
 Raised when no form element plugin with given uid can be found.

exception `fobi.exceptions.FormElementPluginError`
 Bases: `fobi.exceptions.FormPluginError`
 Raised when form element plugin error occurs.

exception `fobi.exceptions.FormHandlerPluginDoesNotExist`
 Bases: `fobi.exceptions.PluginDoesNotExist`
 Raised when no form handler plugin with given uid can be found.

exception `fobi.exceptions.FormHandlerPluginError`
 Bases: `fobi.exceptions.FormPluginError`
 Raised when form handler plugin error occurs.

exception `fobi.exceptions.FormPluginError`
 Bases: `fobi.exceptions.BaseException`
 Base error for form elements and handlers.

exception `fobi.exceptions.FormWizardHandlerPluginDoesNotExist`
 Bases: `fobi.exceptions.PluginDoesNotExist`
 FormWizardHandlerPlugin does not exist.
 Raised when no form wizard handler plugin with given uid can be found.

exception `fobi.exceptions.ImproperlyConfigured`
 Bases: `fobi.exceptions.BaseException`
 Improperly configured.
 Exception raised when developer didn't configure/write the code properly.

exception `fobi.exceptions.IntegrationFormElementPluginDoesNotExist`
 Bases: `fobi.exceptions.PluginDoesNotExist`
 If integration form element plugin with given uid can't be found.

exception `fobi.exceptions.IntegrationFormHandlerPluginDoesNotExist`
 Bases: `fobi.exceptions.PluginDoesNotExist`
 If integration form handler plugin with given uid can't be found.

exception `fobi.exceptions.InvalidRegistryItemType`
 Bases: `ValueError`, `fobi.exceptions.BaseException`
 Invalid registry item type.
 Raised when an attempt is made to register an item in the registry which does not have a proper type.

exception `fobi.exceptions.NoDefaultThemeSet`
 Bases: `fobi.exceptions.ImproperlyConfigured`

Raised when no active theme is chosen.

exception `fobi.exceptions.PluginDoesNotExist`
 Bases: `fobi.exceptions.DoesNotExist`

Raised when no plugin with given uid can be found.

exception `fobi.exceptions.ThemeDoesNotExist`
 Bases: `fobi.exceptions.DoesNotExist`

Raised when no theme with given uid can be found.

38.4.16 fobi.form_importers module

class `fobi.form_importers.BaseFormImporter` (`form_entry_cls`, `form_element_entry_cls`,
`form_properties=None`, `form_data=None`)

Bases: `object`

Base importer.

description = `None`

extract_field_properties (`field_data`)
 Extract field properties.

field_properties_mapping = `None`

field_type_prop_name = `None`

fields_mapping = `None`

get_form_data ()
 Get form data.

get_template_names ()
 Get template names.

get_wizard (`request`, `*args`, `**kwargs`)
 Get wizard.

import_data (`form_properties`, `form_data`)
 Import data.

name = `None`

position_prop_name = `None`

templates = `None`

uid = `None`

wizard = `None`

class `fobi.form_importers.FormImporterPluginRegistry`
 Bases: `fobi.base.BaseRegistry`

Form importer plugins registry.

type
 alias of `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.
```

38.4.17 fobi.form_utils module

```
class fobi.form_utils.ErrorDict
    Bases: django.forms.utils.ErrorDict

    A better ErrorDict.

    as_text()
        As text.

class fobi.form_utils.ErrorList (initlist=None, error_class=None)
    Bases: django.forms.utils.ErrorList

    A better ErrorList.

    as_text()
        As text.
```

38.4.18 fobi.forms module

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

    Bulk change form element plugins form.

    class Meta
        Bases: object

        Meta class.

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

        model
            alias of fobi.models.FormElement

        base_fields = {'groups': <django.forms.models.ModelMultipleChoiceField object>, 'groups_action': <django.forms.models.ModelMultipleChoiceField object>}
        declared_fields = {'groups_action': <django.forms.fields.ChoiceField object>, 'users': <django.forms.fields.ChoiceField object>, 'users_action': <django.forms.fields.ChoiceField object>}}
        media

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

    Bulk change form handler plugins form.

    class Meta
        Bases: object

        Meta class.

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



```
        fields = ('name', 'title', 'is_public', 'active_date_from', 'active_date_to', 'inactive_date_to')
    model
        alias of fobi.models.ModelForm

    base_fields = {'action': <django.forms.fields.CharField object>, 'active_date_from': <django.forms.fields.CharField object>, 'inactive_date_to': <django.forms.fields.CharField object>}

    clean_action()
        Validate the action (URL).

        Checks if URL exists.

    declared_fields = {}

    media

class fobi.forms.ModelFormEntryForm(*args, **kwargs)
    Bases: django.forms.models.ModelForm
    Form for fobi.models.ModelFormEntry model.

    class Meta
        Bases: object
        Meta class.

        fields = ('name',)

    model
        alias of fobi.models.ModelFormEntry

    base_fields = {'name': <django.forms.fields.CharField object>}

    declared_fields = {}

    media

class fobi.forms.ModelFormHandlerEntryForm(data=None, files=None, auto_id='id_%s', prefix=None, initial=None, error_class=<class 'django.forms.utils.ErrorList'>, label_suffix=None, empty_permitted=False, instance=None, use_required_attribute=None)
    Bases: django.forms.models.ModelForm
    FormHandlerEntry form.

    class Meta
        Bases: object
        Meta class.

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

    model
        alias of fobi.models.ModelFormHandlerEntry

    base_fields = {'form_entry': <django.forms.models.ModelChoiceField object>, 'plugin_data': <django.forms.fields.CharField object>, 'plugin_uid': <django.forms.fields.CharField object>}

    declared_fields = {'plugin_uid': <django.forms.fields.ChoiceField object>}

    media
```

```
class fobi.forms.FormHandlerForm(data=None,      files=None,      auto_id='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

FormHandler form.

```
class Meta
```

Bases: object

Meta class.

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

```
model
```

alias of *fobi.models.FormHandler*

```
base_fields = {'groups': <django.forms.models.ModelMultipleChoiceField object>, 'user
```

```
declared_fields = {}
```

```
media
```

```
class fobi.forms.FormWizardEntryForm(*args, **kwargs)
```

Bases: django.forms.models.ModelForm

Form for fobi.models.FormWizardEntry model.

```
class Meta
```

Bases: object

Meta class.

```
fields = ('name', 'title', 'is_public', 'success_page_title', 'success_page_message
```

```
model
```

alias of *fobi.models.FormWizardEntry*

```
base_fields = {'is_public': <django.forms.fields.BooleanField object>, 'name': <djan
```

```
declared_fields = {}
```

```
media
```

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

Bases: django.forms.models.ModelForm

FormWizardFormEntryForm form.

```
class Meta
```

Bases: object

Meta class.

```
fields = ('form_wizard_entry', 'form_entry')
```

```
model
```

alias of *fobi.models.FormWizardFormEntry*

```
base_fields = {'form_entry': <django.forms.models.ModelChoiceField object>, 'form_wiz
```

```

    declared_fields = {}

    media

fobi.forms.FormWizardFormEntryFormSet
    alias of django.forms.formsets.FormWizardFormEntryFormFormSet

class fobi.forms.FormWizardHandlerEntryForm(data=None, files=None,
                                             auto_id='id_%s', prefix=None, initial=None,
                                             error_class=<class 'django.forms.utils.ErrorList'>,
                                             label_suffix=None,
                                             empty_permitted=False, instance=None,
                                             use_required_attribute=None)

Bases: django.forms.models.ModelForm
FormWizardHandlerEntry form.

class Meta
    Bases: object
    Meta class.

    fields = ('form_wizard_entry', 'plugin_data', 'plugin_uid')

    model
        alias of fobi.models.FormWizardHandlerEntry

    base_fields = {'form_wizard_entry': <django.forms.models.ModelChoiceField object>, 'p
    declared_fields = {'plugin_uid': <django.forms.fields.ChoiceField object>}

    media

class fobi.forms.ImportFormEntryForm(data=None, files=None, auto_id='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 = {'file': <django.forms.fields.FileField object>}

    declared_fields = {'file': <django.forms.fields.FileField object>}

    media

class fobi.forms.ImportFormWizardEntryForm(data=None, files=None,
                                             auto_id='id_%s', prefix=None, initial=None,
                                             error_class=<class 'django.forms.utils.ErrorList'>,
                                             label_suffix=None,
                                             empty_permitted=False, field_order=None,
                                             use_required_attribute=None)

Bases: fobi.forms.ImportFormEntryForm
Import form entry wizard form.

    base_fields = {'file': <django.forms.fields.FileField object>}

    declared_fields = {'file': <django.forms.fields.FileField object>}

    media

```

38.4.19 fobi.helpers module

Helpers module. This module can be safely imported from any fobi (sub)module, since it never imports from any of the fobi (sub)modules (except for the *fobi.constants* and *fobi.exceptions* modules).

`fobi.helpers.admin_change_url` (*app_label*, *module_name*, *object_id*, *extra_path=""*,
url_title=None)

Gets an admin change URL for the object given.

Parameters

- **app_label** (*str*) –
- **module_name** (*str*) –
- **object_id** (*int*) –
- **extra_path** (*str*) –
- **url_title** (*str*) – If given, an HTML a tag is returned with *url_title* as the tag title. If left to None just the URL string is returned.

Return str

`fobi.helpers.clean_dict` (*source*, *keys=[]*, *values=[]*)

Removes given keys and values from dictionary.

Parameters

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

Return dict

`fobi.helpers.clone_file` (*upload_dir*, *source_filename*, *relative_path=True*)

Clones the file.

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

Return string Filename of the cloned file.

`fobi.helpers.combine_dicts` (*headers*, *data*)

Combine dicts.

Takes two dictionaries, assuming one contains a mapping keys to titles and another keys to data. Joins as string and returns a result dict.

`fobi.helpers.delete_file` (*image_file*)

Delete file from disc.

`fobi.helpers.do_slugify` (*val*)

Slugify.

`fobi.helpers.ensure_unique_filename` (*destination*)

Makes sure filenames are never overwritten.

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

Return string

`fobi.helpers.extract_file_path` (*name*)

Extracts the file path.

Parameters **name** (*string*) –

Return string

`fobi.helpers.flatatt_inverse_quotes(attrs)`

Convert a dictionary of attributes to a single string.

The returned string will contain a leading space followed by `key="value"`, XML-style pairs. In the case of a boolean value, the key will appear without a value. It is assumed that the keys do not need to be XML-escaped. If the passed dictionary is empty, then return an empty string.

The result is passed through `'mark_safe'` (by way of `'format_html_join'`).

`fobi.helpers.get_app_label_and_model_name(path)`

Gets `app_label` and `model_name` from the path given.

Parameters `path` (*str*) – Dotted path to the model (without `".model"`, as stored in the Django `ContentType` model).

Return tuple `app_label, model_name`

`fobi.helpers.get_form_element_entries_for_form_wizard_entry(form_wizard_entry)`

Get form element entries for the form wizard entry.

`fobi.helpers.get_ignorable_form_values()`

Get ignorable for form values.

Gets an iterable of form values to ignore.

Return iterable

`fobi.helpers.get_model_name_for_object(obj)`

Get model name for object.

Django version agnostic.

`fobi.helpers.get_registered_models(ignore=[])`

Gets registered models as list.

Parameters `ignore` (*iterable*) – Ignore the following content types (should be in `app_label.model` format (example `auth.User`)).

Return list

`fobi.helpers.get_select_field_choices(raw_choices_data, key_type=None, value_type=None, fail_silently=True)`

Get select field choices.

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

Parameters

- **raw_choices_data** (*str*) –
- **key_type** (*type*) –
- **value_type** (*type*) –
- **fail_silently** (*bool*) –

Return list

`fobi.helpers.get_wizard_form_field_value_from_post(request, wizard_view_name, form_key, field_name, fail_silently=True)`

Get wizard form field value from POST.

This is what we could have:


```
>>> request.POST
>>> {
>>>     'csrfmiddlewaretoken': ['kEprTL218a8HNcC02QefNNnF'],
>>>     'slider-form-test_slider': ['14'],
>>>     'form_wizard_view-current_step': ['slider-form'],
>>>     'slider-form-test_email': ['user@example.com']
>>> }
```

Note, that we know nothing about the types here, type conversion should be done manually. The values returned are strings always.

Parameters

- **request** (*django.http.HttpRequest*) –
- **wizard_view_name** (*str*) –
- **form_key** (*str*) – Typically, this would be the step name (form slug).
- **field_name** (*str*) – Field name.
- **fail_silently** (*bool*) – If set to True, no errors raised.

Return str Since everything in session is stored as string.

```
fobi.helpers.get_wizard_form_field_value_from_request(request, wizard_view_name,
                                                         form_key, field_name,
                                                         fail_silently=True, session_priority=False)
```

Get wizard form field value from request.

Note, that we know nothing about the types here, type conversion should be done manually. The values returned are strings always.

Parameters

- **request** (*django.http.HttpRequest*) –
- **wizard_view_name** (*str*) –
- **form_key** (*str*) – Typically, this would be the step name (form slug).
- **field_name** (*str*) – Field name.
- **fail_silently** (*bool*) – If set to True, no errors raised.
- **session_priority** (*bool*) – If set to True, first try to read from session.

Return str Since everything in session is stored as string.

```
fobi.helpers.get_wizard_form_field_value_from_session(request, wizard_view_name,
                                                         form_key, field_name,
                                                         fail_silently=True)
```

Get wizard form field value from session.

This is what we could have:

```
>>> request.session['wizard_form_wizard_view']['step_data']
>>> {
>>>     'slider-form': {
>>>         'csrfmiddlewaretoken': ['DhINThGTgQ50e21DnGG4nYrG0a'],
>>>         'slider-form-test_slider': ['14'],
>>>         'form_wizard_view-current_step': ['slider-form'],
>>>         'slider-form-test_email': ['user@example.com']
>>>     }
```

(continues on next page)

(continued from previous page)

```
>>> }
>>> }
```

Note, that we know nothing about the types here, type conversion should be done manually. The values returned are strings always.

Parameters

- **request** (*django.http.HttpRequest*) –
- **wizard_view_name** (*str*) –
- **form_key** (*str*) – Typically, this would be the step name (form slug).
- **field_name** (*str*) – Field name.
- **fail_silently** (*bool*) – If set to True, no errors raised.

Return str Since everything in session is stored as string.

`fobi.helpers.handle_uploaded_file(upload_dir, image_file)`

Handle uploaded files.

Parameters **image_file** (*django.core.files.uploadedfile.InMemoryUploadedFile*) –

Return string Path to the image (relative).

`fobi.helpers.iterable_to_dict(items, key_attr_name)`

Converts iterable of certain objects to dict.

Parameters

- **items** (*iterable*) –
- **key_attr_name** (*string*) – Attribute to use as a dictionary key.

Return dict

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

Bases: object

Exporting the data into JSON.

export()
Export.

export_to_json()
Export data to JSON.

`fobi.helpers.lists_overlap(sub, main)`

Check whether lists overlap.

`fobi.helpers.map_field_name_to_label(form)`

Takes a form and creates label to field name map.

Parameters **form** (*django.forms.Form*) – Instance of `django.forms.Form`.

Return dict

`fobi.helpers.safe_text(text)`

Safe text (encode).

Return str

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.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.

`fobi.helpers.two_dicts_to_string(headers, data, html_element='p')`

Two dicts to string.

Takes two dictionaries, assuming one contains a mapping keys to titles and another keys to data. Joins as string and returns wrapped into HTML “p” tag.

`fobi.helpers.uniquify_sequence(sequence)`

Uniquify sequence.

Makes sure items in the given sequence are unique, having the original order preserved.

Parameters `sequence` (*iterable*) –

Return list

`fobi.helpers.update_plugin_data(entry, request=None)`

Update plugin data.

Update plugin data of a given entry.

`fobi.helpers.validate_initial_for_choices(plugin_form, field_name_choices='choices', field_name_initial='initial')`

Validate init for choices. Validates the initial value for the choices given.

Parameters

- `plugin_form` (`fobi.base.BaseFormFieldPluginForm`) –
- `field_name_choices` (*str*) –
- `field_name_initial` (*str*) –

Return str

`fobi.helpers.validate_initial_for_multiple_choices(plugin_form, field_name_choices='choices', field_name_initial='initial')`

Validates the initial value for the multiple choices given.

Parameters

- `plugin_form` (`fobi.base.BaseFormFieldPluginForm`) –
- `field_name_choices` (*str*) –
- `field_name_initial` (*str*) –

Return str

`fobi.helpers.validate_submit_value_as(value)`

Validates the `SUBMIT_AS_VALUE`.

Parameters `value` (*str*) –

38.4.20 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

Bases: object

Meta class.

abstract = False

get_registered_plugins()

Get registered plugins.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

groups_list()

Groups list.

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

Return string

plugin_uid_admin()

Plugin uid admin.

Mainly used in admin.

plugin_uid_code()

Plugin uid code.

Mainly used in admin.

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

users_list()

Users list.

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

Return string

class fobi.models.**FormElement** (*args, **kwargs)

Bases: *fobi.models.AbstractPluginModel*

Form element.

Form field plugin. Used when `fobi.settings.RESTRICT_PLUGIN_ACCESS` is set to `True`.

Properties

- *plugin_uid* (str): Plugin UID.
- *users* (django.contrib.auth.models.User): White list of the users allowed to use the form element plugin.
- *groups* (django.contrib.auth.models.Group): White list of the user groups allowed to use the form element plugin.

exception **DoesNotExist**

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception **MultipleObjectsReturned**

Bases: `django.core.exceptions.MultipleObjectsReturned`

get_registered_plugins()

Add choices.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

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

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

plugin_uid

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

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class fobi.models.**FormHandler** (*args, **kwargs)

Bases: *fobi.models.AbstractPluginModel*

Form handler plugin. Used when `fobi.settings.RESTRICT_PLUGIN_ACCESS` is set to `True`.

Properties

- `plugin_uid` (str): Plugin UID.
- `users` (django.contrib.auth.models.User): White list of the users allowed to use the form handler plugin.
- `groups` (django.contrib.auth.models.Group): White list of the user groups allowed to use the form handler plugin.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

get_registered_plugins()

Add choices.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

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

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

plugin_uid

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

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

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

Bases: `fobi.models.AbstractPluginModel`

Form wizard handler plugin. Used when `fobi.settings.RESTRICT_PLUGIN_ACCESS` is set to `True`.

Properties

- `plugin_uid` (str): Plugin UID.

- *users* (django.contrib.auth.models.User): White list of the users allowed to use the form handler plugin.
- *groups* (django.contrib.auth.models.Group): White list of the user groups allowed to use the form handler plugin.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

get_registered_plugins()

Add choices.

groups

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

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

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

plugin_uid

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

users

Accessor to the related objects manager on the forward and reverse sides of a many-to-many relation.

In the example:

```
class Pizza(Model):
    toppings = ManyToManyField(Topping, related_name='pizzas')
```

`pizza.toppings` and `topping.pizzas` are `ManyToManyDescriptor` instances.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

class fobi.models.**AbstractFormWizardPluginEntry** (*args, **kwargs)

Bases: *fobi.models.BaseAbstractPluginEntry*

Abstract form wizard plugin entry.

Properties

- *form_entry* (fobi.models.FormWizardEntry): FormWizard to which the plugin belongs to.
- *plugin_uid* (str): Plugin UID.
- *plugin_data* (str): JSON formatted string with plugin data.


```
class Meta
    Bases: object

    Meta class.

    abstract = False
```

```
entry_user
    Get user.
```

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

    In the example:
```

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

child.parent is a ForwardManyToOneDescriptor instance.

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

```
class fobi.models.AbstractPluginEntry(*args, **kwargs)
    Bases: fobi.models.BaseAbstractPluginEntry
```

Abstract plugin entry.

Properties

- *form_entry* (fobi.models.FormEntry): Form to which the field plugin belongs to.
- *plugin_uid* (str): Plugin UID.
- *plugin_data* (str): JSON formatted string with plugin data.

```
class Meta
    Bases: object

    Meta class.

    abstract = False
```

```
entry_user
    Get user.
```

```
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.

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

```
class fobi.models.BaseAbstractPluginEntry(*args, **kwargs)
    Bases: django.db.models.base.Model
```

Base for AbstractPluginEntry.

Properties

- *plugin_data* (str): JSON formatted string with plugin data.

class Meta

Bases: object

Meta class.

abstract = False

entry_user

Get user from the parent container.

get_plugin (fetch_related_data=False, request=None)

Get plugin.

Gets the plugin class (by `plugin_uid` property), makes an instance of it, serves the data stored in `plugin_data` field (if available). Once all is done, plugin is ready to be rendered.

Parameters `fetch_related_data` (bool) – When set to True, plugin is told to re-fetch all related data (stored in models or other sources).

Return `fobi.base.BasePlugin` Subclass of `fobi.base.BasePlugin`.

get_registered_plugins ()

Get registered plugins.

get_registry ()

Get registry.

plugin_data

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

plugin_name ()

Plugin name.

plugin_uid_code ()

Plugin uid code.

Mainly used in admin.

class fobi.models.FormElementEntry (*args, **kwargs)

Bases: `fobi.models.AbstractPluginEntry`

Form field entry.

Properties

- *form* (`fobi.models.FormEntry`): Form to which the field plugin belongs to.
- *plugin_uid* (str): Plugin UID.
- *plugin_data* (str): JSON formatted string with plugin data.
- *form_fieldset_entry*: Fieldset.
- *position* (int): Entry position.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_entry

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

In the example:

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

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

form_fieldset_entry

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

In the example:

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

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

form_fieldset_entry_id

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

get_registered_plugins()

Gets registered plugins.

get_registry()

Get registry.

id

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

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

plugin_uid

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

position

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

class fobi.models.FormEntry(*args, **kwargs)

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

Form entry.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

action

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

active_date_from

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

active_date_to

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

created

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

formelemententry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

formfieldsetentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

formhandlerentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

formwizardformentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_absolute_url()

Get absolute URL.

Absolute URL, which goes to the form-entry view view page.

Return string

id

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

inactive_page_message

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

inactive_page_title

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

is_active

is_cloneable

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

is_public

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

name

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

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

savedformdataentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

parent.children is a ReverseManyToOneDescriptor instance.

Most of the implementation is delegated to a dynamically defined manager class built by create_forward_many_to_many_manager() defined below.

slug

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

success_page_message

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

success_page_title

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

title

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

updated

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

user

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

In the example:

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

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

user_id

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

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

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

Form fieldset entry.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_entry

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

In the example:

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

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

form_entry_id

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

formelemententry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

id

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

is_repeatable

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

name

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

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

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

Bases: *fobi.models.AbstractPluginEntry*

Form handler entry.

Properties

- *form_entry* (fobi.models.FormEntry): Form to which the handler plugin belongs to.
- *plugin_uid* (str): Plugin UID.
- *plugin_data* (str): JSON formatted string with plugin data.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

form_entry

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

In the example:

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

child.parent is a ForwardManyToOneDescriptor instance.

get_registered_plugins()

Gets registered plugins.

get_registry()

Get registry.

id

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

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

plugin_uid

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

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

Bases: django.db.models.base.Model

Form wizard entry.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

created

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

formwizardformentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

formwizardhandlerentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

get_absolute_url()

Get absolute URL.

Absolute URL, which goes to the form-wizard view view.

Return string

get_wizard_type_display(morekwargs)****id**

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

is_cloneable

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

is_public

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

name

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

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

savewizarddataentry_set

Accessor to the related objects manager on the reverse side of a many-to-one relation.

In the example:

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

`parent.children` is a `ReverseManyToOneDescriptor` instance.

Most of the implementation is delegated to a dynamically defined manager class built by `create_forward_many_to_many_manager()` defined below.

show_all_navigation_buttons

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

slug

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

success_page_message

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

success_page_title

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

title

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

updated

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

user

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

In the example:

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

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

user_id

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

wizard_type

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

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

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

Form wizard form entry.

A coupling point between *FormWizardEntry* and *FormEntry*.

exception DoesNotExist

Bases: `django.core.exceptions.ObjectDoesNotExist`

exception MultipleObjectsReturned

Bases: `django.core.exceptions.MultipleObjectsReturned`

form_entry

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

In the example:

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

child.parent is a ForwardManyToOneDescriptor instance.

form_entry_id

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

form_wizard_entry

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

In the example:

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

child.parent is a ForwardManyToOneDescriptor instance.

form_wizard_entry_id

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

id

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

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

position

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

class fobi.models.**FormWizardHandlerEntry**(*args, **kwargs)

Bases: *fobi.models.AbstractFormWizardPluginEntry*

Form wizard handler entry.

Properties

- *form_wizard_entry* (fobi.models.FormWizardEntry): FormWizard to which the handler plugin belongs to.
- *plugin_uid* (str): Plugin UID.
- *plugin_data* (str): JSON formatted string with plugin data.

exception DoesNotExist

Bases: django.core.exceptions.ObjectDoesNotExist

exception MultipleObjectsReturned

Bases: django.core.exceptions.MultipleObjectsReturned

form_wizard_entry

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

In the example:

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

child.parent is a ForwardManyToOneDescriptor instance.

get_registered_plugins()

Gets registered plugins.

get_registry()

Get registry.

id

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

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

plugin_uid

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

38.4.21 fobi.settings module

- *RESTRICT_PLUGIN_ACCESS* (bool): If set to True, (Django) permission system for fobi plugins is enabled.
- *FORM_ELEMENT_PLUGINS_MODULE_NAME* (str): Name of the module to placed in the (external) apps in which the fobi form element plugin code should be implemented and registered.
- *FORM_HANDLER_PLUGINS_MODULE_NAME* (str): Name of the module to placed in the (external) apps in which the fobi form handler plugin code should be implemented and registered.
- *FORM_CALLBACKS_MODULE_NAME* (str): Name of the module to placed in the (external) apps in which the fobi form callback code should be implemented and registered.
- *FORM_HANDLER_PLUGINS_EXECUTION_ORDER* (tuple): Order in which the form handler plugins are to be executed.
- *FORM_WIZARD_HANDLER_PLUGINS_EXECUTION_ORDER* (tuple): Order in which the form handler plugins are to be executed.
- *DEBUG*

38.4.22 fobi.test module

38.4.23 fobi.utils module

Another helper module. This module can NOT be safely imported from any fobi (sub)module - thus should be imported carefully.

fobi.utils.append_edit_and_delete_links_to_field(*form_element_plugin*,
form_element_entry, *origin=None*,
extra={}, *widget_cls=None*)

Append edit and delete links to the field.

Should return dictionary, which would be used to update default kwargs of form fields.

The hidden inputs *form-{counter}-position* and *form-{counter}-id* are for saving the ordering of the elements (*position* field).

Return dict

fobi.utils.get_allowed_form_element_plugin_uids(*user*)

Get allowed form element plugin uids.

`fobi.utils.get_allowed_form_handler_plugin_uids(user)`

Get allowed form handler plugin uids.

`fobi.utils.get_allowed_form_wizard_handler_plugin_uids(user)`

Get allowed form wizard handler plugin uids.

`fobi.utils.get_allowed_plugin_uids(plugin_model_cls, user)`

Get allowed plugins uids for user given.

Parameters

- **plugin_model_cls** (`fobi.models.AbstractPluginModel`) – Subclass of `fobi.models.AbstractPluginModel`.
- **user** (`django.contrib.auth.models.User`) –

Return list

`fobi.utils.get_user_form_element_plugin_uids(user)`

Get user form element plugin uids.

`fobi.utils.get_user_form_element_plugins(user)`

Get user form element plugins.

`fobi.utils.get_user_form_element_plugins_grouped(user, sort_by_value=False)`

Get user form element plugins grouped.

`fobi.utils.get_user_form_field_plugin_uids(user)`

Get user form element plugin uids.

`fobi.utils.get_user_form_handler_plugin_uids(user)`

Get user form handler plugin uids.

`fobi.utils.get_user_form_handler_plugins(user, exclude_used_singles=False, used_form_handler_plugin_uids=[])`

Get list of plugins allowed for user.

Parameters

- **user** (`django.contrib.auth.models.User`) –
- **exclude_used_singles** (`bool`) –
- **used_form_handler_plugin_uids** (`list`) –

Return list

`fobi.utils.get_user_form_handler_plugins_grouped(user, sort_by_value=False)`

Get user form handler plugins grouped.

`fobi.utils.get_user_form_wizard_handler_plugin_uids(user)`

Get user form handler plugin uids.

`fobi.utils.get_user_form_wizard_handler_plugins(user, exclude_used_singles=False, used_form_wizard_handler_plugin_uids=[])`

Get list of plugins allowed for user.

Parameters

- **user** (`django.contrib.auth.models.User`) –
- **exclude_used_singles** (`bool`) –
- **used_form_wizard_handler_plugin_uids** (`list`) –

Return list

`fobi.utils.get_user_form_wizard_handler_plugins_grouped(user)`

Get user form wizard handler plugins grouped.

`fobi.utils.get_user_plugin_uids(get_allowed_plugin_uids_func,
get_registered_plugin_uids_func, registry, user)`

Gets a list of user plugin uids as a list.

If not yet auto-discovered, auto-discovers them.

Parameters

- **get_allowed_plugin_uids_func** (*callable*) –
- **get_registered_plugin_uids_func** (*callable*) –
- **registry** (`fobi.base.BaseRegistry`) – Subclass of `fobi.base.BaseRegistry` instance.
- **user** (`django.contrib.auth.models.User`) –

Return list

`fobi.utils.get_user_plugins(get_allowed_plugin_uids_func, get_registered_plugins_func, registry, user)`

Get user plugins.

Gets a list of user plugins in a form if tuple (plugin name, plugin description). If not yet autodiscovered, 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_plugins_grouped(get_allowed_plugin_uids_func,
get_registered_plugins_grouped_func, registry, user,
sort_items=True, sort_by_value=False)`

Get user plugins grouped.

Parameters

- **get_allowed_plugin_uids_func** (*callable*) –
- **get_registered_plugins_grouped_func** (*callable*) –
- **registry** (`fobi.base.BaseRegistry`) – Subclass of `fobi.base.BaseRegistry` instance.
- **user** (`django.contrib.auth.models.User`) –
- **sort_items** (*bool*) –
- **sort_by_value** (*bool*) –

Return dict

`fobi.utils.get_wizard_files_upload_dir()`

Get absolute path to the upload directory of fobi form wizard files.

If `WIZARD_FILES_UPLOAD_DIR` path is absolute, return as is. Otherwise, prepend `BASE_PATH`.

Return str Absolute path.

`fobi.utils.perform_form_entry_import(request, form_data)`
Perform form entry import.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_data** (*dict*) –

:return :class:`fobi.modes.FormEntry`: Instance of.

`fobi.utils.prepare_form_entry_export_data(form_entry, form_element_entries=None, form_handler_entries=None)`

Prepare form entry export data.

Parameters

- **form_entry** (*fobi.modes.FormEntry*) – Instance of.
- **form_element_entries** (*django.db.models.QuerySet*) – QuerySet of FormElementEntry instances.
- **form_handler_entries** (*django.db.models.QuerySet*) – QuerySet of FormHandlerEntry instances.

Return str

`fobi.utils.sync_plugins()`
Sync registered plugins.

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

38.4.24 fobi.validators module

`fobi.validators.url_exists(url, local=False)`
Check if URL exists.

Parameters

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

Return bool

38.4.25 fobi.views module

Views.

`fobi.views.add_form_element_entry(request, form_entry_id, form_element_plugin_uid, theme=None, template_name=None)`

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.add_form_handler_entry(request, form_entry_id, form_handler_plugin_uid,
                                theme=None, template_name=None)
```

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.add_form_wizard_form_entry(request, form_wizard_entry_id, form_entry_id,
                                     theme=None, template_name=None)
```

Add form wizard form entry.

Parameters

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

Return django.http.HttpResponse

```
fobi.views.add_form_wizard_handler_entry(request, form_wizard_entry_id,
                                         form_wizard_handler_plugin_uid, theme=None,
                                         template_name=None)
```

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.create_form_entry(request, theme=None, template_name=None)
```

Create form entry.

Parameters

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

- **template_name** (*str*) –

Return `django.http.HttpResponse`

`fobi.views.create_form_wizard_entry` (*request*, *theme=None*, *template_name=None*)

Create form wizard entry.

Parameters

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

Return `django.http.HttpResponse`

`fobi.views.dashboard` (*request*, *theme=None*, *template_name=None*)

Dashboard.

Parameters

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

Return `django.http.HttpResponse`

`fobi.views.delete_form_element_entry` (*request*, *form_element_entry_id*)

Delete form element entry.

Parameters

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

Return `django.http.HttpResponse`

`fobi.views.delete_form_entry` (*request*, *form_entry_id*, *template_name=None*)

Delete form entry.

Parameters

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

Return `django.http.HttpResponse`

`fobi.views.delete_form_handler_entry` (*request*, *form_handler_entry_id*)

Delete form handler entry.

Parameters

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

Return `django.http.HttpResponse`

`fobi.views.delete_form_wizard_entry` (*request*, *form_wizard_entry_id*, *template_name=None*)

Delete form wizard entry.

Parameters

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

Return *django.http.HttpResponse*

fobi.views.delete_form_wizard_form_entry (*request*, *form_wizard_form_entry_id*)
Delete form wizard form entry.

Parameters

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

Return *django.http.HttpResponse*

fobi.views.edit_form_element_entry (*request*, *form_element_entry_id*, *theme=None*, *template_name=None*)
Edit form element entry.

Parameters

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

Return *django.http.HttpResponse*

fobi.views.edit_form_entry (*request*, *form_entry_id*, *theme=None*, *template_name=None*)
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.edit_form_handler_entry (*request*, *form_handler_entry_id*, *theme=None*, *template_name=None*)
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.edit_form_wizard_handler_entry (*request*, *form_wizard_handler_entry_id*, *theme=None*, *template_name=None*)
Edit form handler entry.

Parameters

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

Return *django.http.HttpResponse*

`fobi.views.export_form_entry(request, form_entry_id, template_name=None)`
Export form entry to JSON.

Parameters

- **request** (*django.http.HttpRequest*) –
- **form_entry_id** (*int*) –
- **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.form_importer(request, form_importer_plugin_uid, template_name=None, *args, **kwargs)`
Form importer.

Parameters

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

`fobi.views.form_wizard_entry_submitted(request, form_wizard_entry_slug=None, template_name=None)`
Form wizard entry submitted.

Parameters

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

Return *django.http.HttpResponse*

`fobi.views.form_wizards_dashboard(request, theme=None, template_name=None)`
Dashboard for form wizards.

Parameters

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

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

Return `django.http.HttpResponse`

```
class fobi.views.FormWizardView (**kwargs)
    Bases: fobi.wizard.views.dynamic.DynamicSessionWizardView

    Dynamic form wizard.

    done (form_list, **kwargs)
        Done.

    file_storage = <django.core.files.storage.FileSystemStorage object>

    get_context_data (form, **kwargs)
        Get context data.

    get_form_entry_for_step (step)
        Get form entry title for step.

    get_ignorable_field_names (form_element_entries)
        Get ignorable field names.

    get_initial_wizard_data (request, *args, **kwargs)
        Get initial wizard data.

    post (*args, **kwargs)
        POST requests.

        This method handles POST requests.

        The wizard will render either the current step (if form validation wasn't successful), the next step (if the
        current step was stored successful) or the done view (if no more steps are available)

    render_done (form, **kwargs)
        Render done.

        This method gets called when all forms passed. The method should also re-validate all steps to prevent
        manipulation. If any form fails to validate, render_revalidation_failure should get called. If everything is
        fine call done.
```

```
fobi.views.import_form_entry (request, template_name=None)
    Import form entry.
```

Parameters

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

Return `django.http.HttpResponse`

```
fobi.views.import_form_wizard_entry (request, template_name=None)
    Import form wizard entry.
```

Parameters

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

38.4.26 fobi.widgets module

class `fobi.widgets.BooleanRadioSelect` (*args, **kwargs)

Bases: `django.forms.widgets.RadioSelect`

Boolean radio select for Django.

Example

```
>>> class DummyForm(forms.Form):
>>>     agree = forms.BooleanField(label=_("Agree?"),
>>>                               required=False,
>>>                               widget=BooleanRadioSelect)
```

media

class `fobi.widgets.NumberInput` (attrs=None)

Bases: `django.forms.widgets.TextInput`

input_type = 'number'

media

class `fobi.widgets.RichSelect` (attrs=None, choices=(), prepend_html=None, append_html=None, override_name=None)

Bases: `django.forms.widgets.Select`

Rich select widget with some rich enhancements.

Based on original Select widget and intended to be a drop-off replacement.

media

render (name, value, attrs=None, **kwargs)

Renders the element, having prepended and appended extra parts.

class `fobi.widgets.RichSelectInverseQuotes` (attrs=None, choices=(), prepend_html=None, append_html=None, override_name=None)

Bases: `fobi.widgets.RichSelect`

Almost same as original, but uses alternative flatatt function.

Uses inverse quotes.

media

option_template_name = 'fobi/django/forms/widgets/rich_select_inverse_option.html'

template_name = 'fobi/django/forms/widgets/rich_select_inverse.html'

38.4.27 Module contents

CHAPTER 39

Indices and tables

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

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

40.1 `fobi.contrib.apps.djangocms_integration`

A `django-fobi` integration with DjangoCMS.

40.1.1 Prerequisites

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

40.1.2 Installation

40.1.2.1 Versions

See the requirements files:

- [DjangoCMS 2.4.3 requirements](#).
- [DjangoCMS 3.0.6 requirements](#).
- [DjangoCMS 3.4.3 requirements](#).

40.1.2.2 `your_project/settings.py`

See the example settings files:

- [DjangoCMS 2.4.3 settings](#).
- [DjangoCMS 3.0.6/3.4.3. settings](#).

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

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

40.1.3 Information for developers

40.1.3.1 Templates for DjangoCMS

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

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

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

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

40.1.3.2 Template rendering

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

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

40.1.3.3 Using custom templates for rendering the widget

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

Example:

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

Same goes for form-sent templates.

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


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

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

40.1.4 Usage

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

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

40.2 fobi.contrib.apps.drf_integration

A django-fobi integration with Django REST framework.

Supported actions are:

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

40.2.1 Live demo

Live demo is available on Heroku.

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

40.2.2 Supported fields

The following fields are supported.

40.2.2.1 Content (presentational form elements)

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

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

40.2.2.2 Fields

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

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

40.2.3 Not (yet) supported fields

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

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

40.2.4 Implementation details

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

Some of the plugins may seem to have zero-added-value and in fact they are. For instance, DRF integration `slider` plugin is just an exact copy of the `range_select` plugin, created in order to provide exactly the same form fields generated in the API.

You should mention all the plugins you want to use explicitly in the project settings. Thus, if you have used (included in the `INSTALLED_APPS`) the core plugins:

- `fobi.contrib.plugins.form_elements.fields.boolean`
- `fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple`
- `fobi.contrib.plugins.form_elements.fields.date`
- `fobi.contrib.plugins.form_elements.fields.date_drop_down`

- `fobi.contrib.plugins.form_elements.fields.datetime`
- `fobi.contrib.plugins.form_elements.fields.decimal`
- `fobi.contrib.plugins.form_elements.fields.duration`
- `fobi.contrib.plugins.form_elements.fields.email`
- `fobi.contrib.plugins.form_elements.fields.file`
- `fobi.contrib.plugins.form_elements.fields.float`
- `fobi.contrib.plugins.form_elements.fields.hidden`
- `fobi.contrib.plugins.form_elements.fields.input`
- `fobi.contrib.plugins.form_elements.fields.integer`
- `fobi.contrib.plugins.form_elements.fields.ip_address`
- `fobi.contrib.plugins.form_elements.fields.null_boolean`
- `fobi.contrib.plugins.form_elements.fields.password`
- `fobi.contrib.plugins.form_elements.fields.radio`
- `fobi.contrib.plugins.form_elements.fields.range_select`
- `fobi.contrib.plugins.form_elements.fields.regex`
- `fobi.contrib.plugins.form_elements.fields.select`
- `fobi.contrib.plugins.form_elements.fields.select_multiple`
- `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max`
- `fobi.contrib.plugins.form_elements.fields.slider`
- `fobi.contrib.plugins.form_elements.fields.slug`
- `fobi.contrib.plugins.form_elements.fields.text`
- `fobi.contrib.plugins.form_elements.fields.textarea`
- `fobi.contrib.plugins.form_elements.fields.time`
- `fobi.contrib.plugins.form_elements.fields.url`
- `fobi.contrib.plugins.form_elements.content.content_image`
- `fobi.contrib.plugins.form_elements.content.content_image_url`
- `fobi.contrib.plugins.form_elements.content.content_richtext`
- `fobi.contrib.plugins.form_elements.content.content_text`
- `fobi.contrib.plugins.form_elements.content.content_video`
- `fobi.contrib.plugins.form_handlers.db_store`
- `fobi.contrib.plugins.form_handlers.http_repost`
- `fobi.contrib.plugins.form_handlers.mail`

You should include their correspondent Django REST framework implementations in the `INSTALLED_APPS` as well:

- `fobi.contrib.apps.drf_integration.form_elements.fields.boolean`
- `fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple`
- `fobi.contrib.apps.drf_integration.form_elements.fields.date`

- `fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down`
- `fobi.contrib.apps.drf_integration.form_elements.fields.datetime`
- `fobi.contrib.apps.drf_integration.form_elements.fields.decimal`
- `fobi.contrib.apps.drf_integration.form_elements.fields.duration`
- `fobi.contrib.apps.drf_integration.form_elements.fields.email`
- `fobi.contrib.apps.drf_integration.form_elements.fields.file`
- `fobi.contrib.apps.drf_integration.form_elements.fields.float`
- `fobi.contrib.apps.drf_integration.form_elements.fields.hidden`
- `fobi.contrib.apps.drf_integration.form_elements.fields.input`
- `fobi.contrib.apps.drf_integration.form_elements.fields.integer`
- `fobi.contrib.apps.drf_integration.form_elements.fields.ip_address`
- `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean`
- `fobi.contrib.apps.drf_integration.form_elements.fields.password`
- `fobi.contrib.apps.drf_integration.form_elements.fields.radio`
- `fobi.contrib.apps.drf_integration.form_elements.fields.range_select`
- `fobi.contrib.apps.drf_integration.form_elements.fields.regex`
- `fobi.contrib.apps.drf_integration.form_elements.fields.select`
- `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple`
- `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max`
- `fobi.contrib.apps.drf_integration.form_elements.fields.slider`
- `fobi.contrib.apps.drf_integration.form_elements.fields.slug`
- `fobi.contrib.apps.drf_integration.form_elements.fields.text`
- `fobi.contrib.apps.drf_integration.form_elements.fields.textarea`
- `fobi.contrib.apps.drf_integration.form_elements.fields.time`
- `fobi.contrib.apps.drf_integration.form_elements.fields.url`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_image`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_image_url`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_richtext`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_text`
- `fobi.contrib.apps.drf_integration.form_elements.content.content_video`
- `fobi.contrib.apps.drf_integration.form_handlers.db_store`
- `fobi.contrib.apps.drf_integration.form_handlers.http_repost`
- `fobi.contrib.apps.drf_integration.form_handlers.mail`

40.2.5 Installation

40.2.5.1 Versions

Was made with `django-rest-framework` 3.6.2. May work on earlier versions, although not guaranteed.

See the [requirements file](#).

40.2.5.2 `your_project/settings.py`

See the [example settings file](#).

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    # ...
    # Here should come a list of form element plugins of the core
    # package, followed by the list of form handler plugins of the core
    # package, followed by the list of themes of the core package and
    # all other apps that do matter.
    # ...
    'rest_framework', # Django REST framework
    'fobi.contrib.apps.drf_integration', # DRF integration app

    # DRF integration form element plugins - fields
    'fobi.contrib.apps.drf_integration.form_elements.fields.boolean',
    'fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple',
    'fobi.contrib.apps.drf_integration.form_elements.fields.date',
    'fobi.contrib.apps.drf_integration.form_elements.fields.datetime',
    'fobi.contrib.apps.drf_integration.form_elements.fields.decimal',
    'fobi.contrib.apps.drf_integration.form_elements.fields.duration',
    'fobi.contrib.apps.drf_integration.form_elements.fields.email',
    'fobi.contrib.apps.drf_integration.form_elements.fields.file',
    'fobi.contrib.apps.drf_integration.form_elements.fields.float',
    'fobi.contrib.apps.drf_integration.form_elements.fields.hidden',
    'fobi.contrib.apps.drf_integration.form_elements.fields.input',
    'fobi.contrib.apps.drf_integration.form_elements.fields.integer',
    'fobi.contrib.apps.drf_integration.form_elements.fields.ip_address',
    'fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean',
    'fobi.contrib.apps.drf_integration.form_elements.fields.password',
    'fobi.contrib.apps.drf_integration.form_elements.fields.radio',
    'fobi.contrib.apps.drf_integration.form_elements.fields.range_select',
    'fobi.contrib.apps.drf_integration.form_elements.fields.regex',
    'fobi.contrib.apps.drf_integration.form_elements.fields.select',
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple',
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple_with_max',
    'fobi.contrib.apps.drf_integration.form_elements.fields.slider',
    'fobi.contrib.apps.drf_integration.form_elements.fields.slug',
    'fobi.contrib.apps.drf_integration.form_elements.fields.text',
    'fobi.contrib.apps.drf_integration.form_elements.fields.textarea',
    'fobi.contrib.apps.drf_integration.form_elements.fields.time',
    'fobi.contrib.apps.drf_integration.form_elements.fields.url',

    # DRF integration form element plugins - presentational
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image',
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image_url',
    'fobi.contrib.apps.drf_integration.form_elements.content.content_richtext',
    'fobi.contrib.apps.drf_integration.form_elements.content.content_text',
```

(continues on next page)

(continued from previous page)

```
'fobi.contrib.apps.drf_integration.form_elements.content.content_video',

# DRF integration form handler plugins
'fobi.contrib.apps.drf_integration.form_handlers.db_store',
'fobi.contrib.apps.drf_integration.form_handlers.mail',
'fobi.contrib.apps.drf_integration.form_handlers.http_repost',
# ...
]
```

40.2.5.3 your_project/urls.py

Add the following code to the main `urls.py` of your project:

```
# Conditionally including django-rest-framework integration app
if 'fobi.contrib.apps.drf_integration' in settings.INSTALLED_APPS:
    from fobi.contrib.apps.drf_integration.urls import fobi_router
    urlpatterns += [
        url(r'^api/', include(fobi_router.urls))
    ]
```

40.2.6 Usage

If you have followed the steps above precisely, you would be able to access the API using `http://localhost:8000/api/fobi-form-entry/`.

Actions/methods supported:

40.2.6.1 LIST

```
GET /api/fobi-form-entry/
```

Lists all the forms available. Anonymous users would see the list of all public forms. Authenticated users would see their own forms in addition to the public forms.

40.2.6.2 OPTIONS

```
OPTIONS /api/fobi-form-entry/{FORM_SLUG}/
```

Lists all field options for the selected form.

See the [test DRF form](#) and [same form in DRF integration app](#) with most of the fields that do have rich additional metadata.

OPTIONS call produces the following response:

```
OPTIONS /api/fobi-form-entry/test-drf-form/
HTTP 200 OK
Allow: GET, PUT, PATCH, OPTIONS
Content-Type: application/json
Vary: Accept
```

```
{
  "name": "Fobi Form Entry Instance",
  "description": "FormEntry view set.",
  "renders": [
    "application/json",
    "text/html"
  ],
  "parses": [
    "application/json",
    "application/x-www-form-urlencoded",
    "multipart/form-data"
  ],
  "actions": {
    "PUT": {
      "test_integer": {
        "type": "integer",
        "required": false,
        "read_only": false,
        "label": "Test integer",
        "min_value": 1,
        "max_value": 20,
        "initial": 10
      },
      "test_email": {
        "type": "email",
        "required": true,
        "read_only": false,
        "label": "Test email",
        "help_text": "Donec mollis hendrerit risus. Phasellus a "
          "est. Nam ipsum risus, rutrum vitae, "
          "vestibulum eu, molestie vel, lacus. "
          "Praesent nec nisl a purus blandit viverra. "
          "Cras id dui.",
        "max_length": 255,
        "placeholder": "john@doe.com"
      },
      "test_text": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Test text",
        "help_text": "Sed lectus. Phasellus gravida semper "
          "nisi. Curabitur at lacus ac velit ornare "
          "lobortis. Mauris turpis nunc, blandit et, "
          "volutpat molestie, porta ut, ligula. Lorem "
          "ipsum dolor sit amet, consectetur "
          "adipiscing elit.",
        "max_length": 255,
        "placeholder": "Lorem ipsum dolor sit amet"
      },
      "test_url": {
        "type": "url",
        "required": false,
        "read_only": false,
        "label": "Test URL",
        "max_length": 255,
        "initial": "http://github.com"
      }
    }
  }
}
```

(continues on next page)

(continued from previous page)

```

    },
    "test_decimal_field": {
        "type": "decimal",
        "required": false,
        "read_only": false,
        "label": "Test decimal field",
        "min_value": 1.0,
        "max_value": 25.0,
        "initial": 10.0,
        "placeholder": "3.14",
        "max_digits": 5,
        "decimal_places": 2
    },
    "test_float_field": {
        "type": "float",
        "required": false,
        "read_only": false,
        "label": "Test float field",
        "min_value": 1.0,
        "max_value": 10.0,
        "initial": 3.14
    },
    "test_ip_address": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Test IP address",
        "max_length": 255,
        "placeholder": "127.0.0.1"
    },
    "test_password_field": {
        "type": "string",
        "required": false,
        "read_only": false,
        "label": "Test password field",
        "max_length": 255,
        "placeholder": "your-secret-password"
    },
    "test_regex_field": {
        "type": "regex",
        "required": false,
        "read_only": false,
        "label": "Test regex field",
        "max_length": 255,
        "regex": "^[a-zA-Z])+$"
    },
    "test_slug_field": {
        "type": "slug",
        "required": false,
        "read_only": false,
        "label": "Test slug field",
        "max_length": 255,
        "placeholder": "lorem-ipsum-dolor-sit-amet"
    },
    "test_textarea_field": {
        "type": "string",
        "required": false,

```

(continues on next page)

(continued from previous page)

```

        "read_only": false,
        "label": "Test textarea field",
        "placeholder": "Pellentesque habitant morbi tristique."
    },
    "test_input_field": {
        "type": "string",
        "required": false,
        "read_only": true,
        "label": "Test input field",
        "max_length": 255,
        "autofocus": "autofocus",
        "autocomplete": "on",
        "disabled": "disabled"
    },
    "content_image_url_b0996b16-9f1c-430d-a6c7-0a722f4c2177": {
        "type": "content",
        "required": false,
        "read_only": true,
        "initial": "<p><img src=\"http://example.com/image.jpg\" alt=\"n.n.\"
↪width=\"600\"/></p>",
        "contenttype": "image",
        "raw_data": {
            "url": "http://example.com/image.jpg",
            "alt": "n.n.",
            "fit_method": "fit_width",
            "size": "600x600"
        },
        "content": "<p><img src=\"http://example.com/image.jpg\" alt=\"n.n.\"
↪width=\"600\"/></p>"
    },
    "content_text_de4d69b2-99e1-479d-8c61-1534dea7c981": {
        "type": "content",
        "required": false,
        "read_only": true,
        "initial": "<p>Pellentesque posuere. Quisque id mi. "
            "Duis arcu tortor, suscipit eget, imperdiet "
            "nec, imperdiet iaculis, ipsum. Phasellus a "
            "est. In turpis.</p>",
        "contenttype": "text",
        "raw_data": {
            "text": "Pellentesque posuere. Quisque id mi. Duis "
                "arcu tortor, suscipit eget, imperdiet nec, "
                "imperdiet iaculis, ipsum. Phasellus a est. "
                "In turpis."
        },
        "content": "<p>Pellentesque posuere. Quisque id mi. Duis "
            "arcu tortor, suscipit eget, imperdiet nec, "
            "imperdiet iaculis, ipsum. Phasellus a est. "
            "In turpis.</p>"
    },
    "content_video_f4799aca-9a0b-4f1a-8069-dda611858ef4": {
        "type": "content",
        "required": false,
        "read_only": true,
        "initial": "<iframe src=\"//www.youtube.com/embed/8GVIui0JK0M\"
↪width=\"500\" height=\"400\" frameborder=\"0\" allowfullscreen></iframe>",
        "contenttype": "video",

```

(continues on next page)

(continued from previous page)

```

        "raw_data": {
            "title": "Delusional Insanity - To far beyond...",
            "url": "https://www.youtube.com/watch?v=8GVIui0JK0M&t=1s",
            "size": "500x400"
        },
        "content": "<iframe src=\"//www.youtube.com/embed/8GVIui0JK0M\"
↪width=\"500\" height=\"400\" frameborder=\"0\" allowfullscreen></iframe>"
    }
}

```

Some insights:

Meta-data is passed to the `DRFIntegrationFormElementPluginProcessor` as `field_metadata` argument, which is supposed to be a dict.

- Example 1: content_image plugin
- Example 2: decimal plugin
- Example 3: text plugin

Private forms would be only visible to authenticated users.

40.2.6.3 PUT

```

PUT /api/fobi-form-entry/{FORM_SLUG}/

{DATA}

```

40.2.7 Callbacks

Callbacks work just the same way the core callbacks work.

40.2.7.1 fobi_form_callbacks.py

```

from fobi.base import (
    integration_form_callback_registry,
    IntegrationFormCallback,
)

from fobi.constants import (
    CALLBACK_BEFORE_FORM_VALIDATION,
    CALLBACK_FORM_INVALID,
    CALLBACK_FORM_VALID,
    CALLBACK_FORM_VALID_AFTER_FORM_HANDLERS,
    CALLBACK_FORM_VALID_BEFORE_SUBMIT_PLUGIN_FORM_DATA,
)

from fobi.contrib.apps.drf_integration import UID as INTEGRATE_WITH

class DRFSaveAsFooItem(IntegrationFormCallback):

```

(continues on next page)

(continued from previous page)

```
"""Save the form as a foo item, if certain conditions are met."""

stage = CALLBACK_FORM_VALID
integrate_with = INTEGRATE_WITH

def callback(self, form_entry, request, **kwargs):
    """Custom callback login comes here."""
    logger.debug("Great! Your form is valid!")

class DRFDummyInvalidCallback(IntegrationFormCallback):
    """Saves the form as a foo item, if certain conditions are met."""

    stage = CALLBACK_FORM_INVALID
    integrate_with = INTEGRATE_WITH

    def callback(self, form_entry, request, **kwargs):
        """Custom callback login comes here."""
        logger.debug("Damn! You've made a mistake, boy!")
```

40.2.8 Testing

To test Django REST framework integration package only, run the following command:

```
./runtests.py src/fobi/tests/test_drf_integration.py
```

or use plain Django tests:

```
./manage.py test fobi.tests.test_drf_integration --settings=settings.test
```

40.2.9 Limitations

Certain fields are not available yet (relational fields).

40.3 Form elements

40.3.1 Content form element

Presentational form elements for drf_integration.

40.3.1.1 fobi.contrib.apps.drf_integration.form_elements.content.content_image

A django-fobi ContentImage plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentImage`.

40.3.1.1.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_image` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.1.1.2 Usage

Unlike standard fields, `ContentImage` field is purely presentational. You’re not supposed to make write actions on it (it won’t work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
{
  "actions": {
    "PUT": {
      // ...
      "content_image_89c8c319-195b-487a-a44d-f59ef14a5d44": {
        "type": "content",
        "required": false,
        "read_only": true,
        "contenttype": "image",
        "content": "\n<p>\n\n\n\n\n<img src=\"/media/fobi_plugins/content_image/
↪test-image-thumbnail.jpg\" alt=\"Lorem ipsum\"/>\n\n\n</p>\n",
        "raw": {
          "file": "/media/fobi_plugins/content_image/test-image.jpg",
          "alt": "Lorem ipsum",
          "fit_method": "center",
          "size": "500x500"
        }
      },
      // ...
    }
  }
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “image” for `ContentImage` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentImage` plugin. Contains “file”, “alt”, “fit_method” and “size” attributes.

40.3.1.2 fobi.contrib.apps.drf_integration.form_elements.content.content_image_url

A django-fobi `ContentImageURL` plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentImage`.

40.3.1.2.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_image_url` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.content.content_image_url',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.1.2.2 Usage

Unlike standard fields, `ContentImageURL` field is purely presentational. You’re not supposed to make write actions on it (it won’t work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
"actions": {  
    "PUT": {  
        // ...  
        "content_image_89c8c319-195b-487a-a44d-f59ef14a5d44": {  
            "type": "content",  
            "required": false,  
            "read_only": true,  
            "contenttype": "image",  
            "content": "\n<p>\n\n\n\n\n<img src=\"http://example.com/media/test-image.  
→jpg\" alt=\"Lorem ipsum\"/>\n\n\n\n</p>\n",  
            "raw": {  
                "url": "http://example.com/media/test-image.jpg",  
                "alt": "Lorem ipsum",  
                "fit_method": "fit_width",  
                "size": "500x500"  
            }  
        },  
        // ...  
    }  
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “image” for `ContentImageURL` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentImageURL` plugin. Contains “url”, “alt”, “fit_method” and “size” attributes.

40.3.1.3 fobi.contrib.apps.drf_integration.form_elements.content.content_richtext

A django-fobi ContentRichText plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentRichText`.

40.3.1.3.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_richtext` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.content.content_richtext',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.1.3.2 Usage

Unlike standard fields, `ContentText` field is purely presentational. You're not supposed to make write actions on it (it won't work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
{
  "actions": {
    "PUT": {
      // ...
      "content_text_89c8c319-195b-487a-a44d-f59ef14a5d44": {
        "type": "content",
        "required": false,
        "read_only": true,
        "contenttype": "text",
        "content": "\n<p>\n\nLorem ipsum dolor sit amet.\n\n\n</p>\n",
        "raw": {
          "text": "Lorem ipsum dolor sit amet."
        }
      },
      // ...
    }
  }
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “text” for `ContentText` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentText` plugin. Contains “text” attribute.

40.3.1.4 fobi.contrib.apps.drf_integration.form_elements.content.content_text

A django-fobi ContentText plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentText`.

40.3.1.4.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.content.content_text',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.1.4.2 Usage

Unlike standard fields, `ContentText` field is purely presentational. You're not supposed to make write actions on it (it won't work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
{  
  "actions": {  
    "PUT": {  
      // ...  
      "content_text_89c8c319-195b-487a-a44d-f59ef14a5d44": {  
        "type": "content",  
        "required": false,  
        "read_only": true,  
        "contenttype": "text",  
        "content": "\n<p>\n\nLorem ipsum dolor sit amet.\n\n</p>\n",  
        "raw": {  
          "text": "Lorem ipsum dolor sit amet."  
        }  
      },  
      // ...  
    }  
  }  
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “text” for `ContentText` field.
- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentText` plugin. Contains “text” attribute.

40.3.1.5 fobi.contrib.apps.drf_integration.form_elements.content.content_video

A django-fobi ContentVideo plugin for integration with Django REST framework. Makes use of the `fobi.contrib.apps.drf_integration.fields.ContentVideo`.

40.3.1.5.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.content.content_video` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.content.content_video',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.1.5.2 Usage

Unlike standard fields, `ContentVideo` field is purely presentational. You're not supposed to make write actions on it (it won't work). Neither will it be displayed in the browsable API (list/retrieve actions). However, it will be listed in the options action call.

Sample JSON response fragment

```
{
  "actions": {
    "PUT": {
      // ...
      "content_video_41a6b951-e6f9-4f08-ada6-3b109aa9a72f": {
        "type": "content",
        "required": false,
        "read_only": true,
        "contenttype": "video",
        "content": "\n<iframe src=\"//www.youtube.com/embed/3PlqcVcs4Ik\" width=\n
→ "500\" height=\"400\" frameborder=\"0\" allowfullscreen></iframe>\n",
        "raw": {
          "title": "Cras risus ipsum faucibus",
          "url": "https://www.youtube.com/watch?v=3PlqcVcs4Ik",
          "size": "500x400"
        }
      },
      // ...
    }
  }
}
```

JSON response fragment explained

- `type` (str): Set to “content” for all presentational form elements.
- `contenttype` (str): Set to “video” for `ContentVideo` field.

- `content` (str): Representation of the content. Rendered partial HTML.
- `raw` (json dict): Raw attributes of the `ContentVideo` plugin. Contains “title”, “url” and “size” attributes.

40.3.2 Fields

Form fields for `drf_integration`.

40.3.2.1 `fobi.contrib.apps.drf_integration.form_elements.fields.boolean`

A `django-fobi` `BooleanField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.BooleanField`.

40.3.2.1.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.boolean',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.2 `fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple`

A `django-fobi` `CharField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.MultipleChoiceField`.

40.3.2.2.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_  
↪multiple',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.3 fobi.contrib.apps.drf_integration.form_elements.fields.date

A django-fobi DateField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DateField`.

40.3.2.3.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.date` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.date',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.4 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down

A django-fobi DateField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DateField`.

This plugin has been made primarily for compatibility with `date_drop_down` plugin of the core package.

40.3.2.4.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.5 fobi.contrib.apps.drf_integration.form_elements.fields.datetime

A django-fobi DateField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DateTimeField`.

40.3.2.5.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.datetime` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.datetime',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.6 fobi.contrib.apps.drf_integration.form_elements.fields.decimal

A django-fobi DecimalField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.DecimalField`.

40.3.2.6.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.decimal` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.decimal',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.7 fobi.contrib.apps.drf_integration.form_elements.fields.email

A django-fobi EmailField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.EmailField`.

40.3.2.7.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.email` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.email',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.8 fobi.contrib.apps.drf_integration.form_elements.fields.file

A django-fobi `FileField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.FileField`.

40.3.2.8.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.file` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.file',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.9 fobi.contrib.apps.drf_integration.form_elements.fields.float

A django-fobi `FloatField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.FloatField`.

40.3.2.9.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.float` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.float',  
    # ...  
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.10 `fobi.contrib.apps.drf_integration.form_elements.fields.hidden`

A `django-fobi` `HiddenField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.HiddenField`.

Note, that in terms of the Django REST framework it is a read-only field. Any values posted along won't be saved. Initial value would.

40.3.2.10.1 Installation

(1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.hidden` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.hidden',  
    # ...  
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.11 `fobi.contrib.apps.drf_integration.form_elements.fields.input`

A `django-fobi` `CharField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

40.3.2.11.1 Installation

(1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.input` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.input',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.12 fobi.contrib.apps.drf_integration.form_elements.fields.integer

A django-fobi `IntegerField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.IntegerField`.

40.3.2.12.1 Installation

(1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.integer` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.integer',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.13 fobi.contrib.apps.drf_integration.form_elements.fields.ip_address

A django-fobi `IPAddressField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.IPAddressField`.

40.3.2.13.1 Installation

(1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.ip_address` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.ip_address',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.14 `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean`

A django-fobi `NullBooleanField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.NullBooleanField`.

40.3.2.14.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.15 `fobi.contrib.apps.drf_integration.form_elements.fields.password`

A django-fobi `CharField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

40.3.2.15.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.password` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.password',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.16 fobi.contrib.apps.drf_integration.form_elements.fields.radio

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

40.3.2.16.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.radio` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.radio',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.17 fobi.contrib.apps.drf_integration.form_elements.fields.range_select

A django-fobi ChoiceField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

40.3.2.17.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.range_select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.range_select',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.18 fobi.contrib.apps.drf_integration.form_elements.fields.regex

A django-fobi RegexpField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.RegexpField`.

40.3.2.18.1 Installation

1. Add `fobi.contrib.apps.drf_integration.form_elements.fields.regex` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.regex',  
    # ...  
)
```

2. In the terminal type:

```
./manage.py fobi_sync_plugins
```

3. Assign appropriate permissions to the target users/groups to be using the plugin if `FBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.19 `fobi.contrib.apps.drf_integration.form_elements.fields.select`

A `django-fobi` `ChoiceField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

40.3.2.19.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.select',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.20 `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple`

A `django-fobi` `CharField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.MultipleChoiceField`.

40.3.2.20.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.21 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.MultipleChoiceField`.

40.3.2.21.1 Installation

(1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple` to the `INSTALLED_APPS` in your settings.py.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.22 fobi.contrib.apps.drf_integration.form_elements.fields.slider

A django-fobi ChoiceField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.ChoiceField`.

40.3.2.22.1 Installation

(1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.slider` to the `INSTALLED_APPS` in your settings.py.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_elements.fields.slider',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.23 `fobi.contrib.apps.drf_integration.form_elements.fields.slug`

A `django-fobi` `SlugField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.SlugField`.

40.3.2.23.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.slug` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.slug',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.24 `fobi.contrib.apps.drf_integration.form_elements.fields.text`

A `django-fobi` `CharField` plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

40.3.2.24.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.text',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.25 fobi.contrib.apps.drf_integration.form_elements.fields.textarea

A django-fobi CharField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.CharField`.

40.3.2.25.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.textarea` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.textarea',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.26 fobi.contrib.apps.drf_integration.form_elements.fields.time

A django-fobi TimeField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.TimeField`.

40.3.2.26.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.time` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.time',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.3.2.27 fobi.contrib.apps.drf_integration.form_elements.fields.url

A django-fobi URLField plugin for integration with Django REST framework. Makes use of the `rest_framework.fields.URLField`.

40.3.2.27.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_elements.fields.time` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_elements.fields.url',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.4 Form handlers

Form handlers for `drf_integration`.

40.4.1 fobi.contrib.apps.drf_integration.form_handlers.db_store

A django-fobi Mail form handler plugin for integration with Django REST framework. Saves submitted form data into the `SavedFormDataEntry` model.

40.4.1.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_handlers.db_store` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.apps.drf_integration.form_handlers.db_store',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py migrate  
  
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.4.2 fobi.contrib.apps.drf_integration.form_handlers.http_repost

A django-fobi HTTP repost form handler plugin for integration with Django REST framework. Submits the form to another endpoint specified.

40.4.2.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_handlers.http_repost` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_handlers.http_repost',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.4.3 fobi.contrib.apps.drf_integration.form_handlers.mail

A django-fobi Mail form handler plugin for integration with Django REST framework. Submits the form data by email to the specified email address.

40.4.3.1 Installation

- (1) Add `fobi.contrib.apps.drf_integration.form_handlers.mail` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.apps.drf_integration.form_handlers.mail',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py migrate

./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.5 fobi.contrib.apps.feincms_integration

A django-fobi integration with FeinCMS.

40.5.1 Prerequisites

Tested with FeinCMS 1.10, 1.12 and 1.13 only. Might work on earlier (or later) versions as well.

40.5.2 Installation

40.5.2.1 Versions

See the requirements files:

- [FeinCMS 1.10 requirements](#).
- [FeinCMS 1.12 requirements](#).
- [FeinCMS 1.13 requirements](#).

40.5.2.2 `your_project/settings.py`

See the [example settings file](#).

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    'feincms', # FeinCMS

    'fobi.contrib.apps.feincms_integration', # Fobi FeinCMS app

    'page', # Example
]

FEINCMS_RICHTEXT_INIT_CONTEXT = {
    'TINYMCE_JS_URL': STATIC_URL + 'tiny_mce/tiny_mce.js',
}
```

40.5.2.3 `your_project/page/models.py`

```
from django.utils.translation import ugettext_lazy as _

from feincms.module.page.models import Page
from feincms.content.raw.models import RawContent
from feincms.content.richtext.models import RichTextContent
# Import the ``django-fobi`` widget.
from fobi.contrib.apps.feincms_integration.widgets import FobiFormWidget

Page.register_extensions('feincms.module.extensions.translations',)

# Register basic template.
Page.register_templates(
    {
        'title': _(u"Base template"),
        'path': 'page/base.html',
        'key': 'page_base',
        'regions': (
            ('main', _(u"Main")),
            ('sidebar', _(u"Sidebar")),
        )
    },
)

# Standard content types
```

(continues on next page)

(continued from previous page)

```
Page.create_content_type(RawContent)
Page.create_content_type(RichTextContent)

# Register the ``django-fobi`` widget.
Page.create_content_type(FobiFormWidget)
```

40.5.2.4 your_project/admin.py

```
from django.contrib import admin

from feincms.module.page.modeladmins import PageAdmin

from page.models import Page

admin.site.register(Page, PageAdmin)
```

40.5.2.5 Information for developers

40.5.2.5.1 Template rendering

The embed FeinCMS widget is rendered with use of two theme templates:

- `view_embed_form_entry_ajax_template`: Used for rendering the form.
- `embed_form_entry_submitted_ajax_template`: Used for rendering the form sent event.

40.5.2.6 Using custom templates for rendering the widget

In the widget, you can specify a template which you want to be used for rendering the form or the form-sent event.

Example:

```
FOBI_FEINCMS_INTEGRATION_FORM_TEMPLATE_CHOICES = (
    ("yourapp/custom_view_embed_form_v1.html",
     "Custom embed form view template #1"),
    ("yourapp/custom_view_embed_form_v2.html",
     "Custom embed form view template #2"),
)
```

Same goes for form-sent templates.

```
FOBI_FEINCMS_INTEGRATION_SUCCESS_PAGE_TEMPLATE_CHOICES = (
    ("yourapp/custom_embed_form_submitted_v1.html",
     "Custom form-sent template #1"),
    ("yourapp/custom_embed_form_submitted_v2.html",
     "Custom form-sent template #2"),
)
```

Registering a template in the `FORM_TEMPLATE_CHOICES` makes it available for all the themes. If you rather want to use different custom templates for different themes, use the `FOBI_CUSTOM_THEME_DATA` as shown in the example below.

```
FOBI_CUSTOM_THEME_DATA = {
    'bootstrap3': {
        'feincms_integration': {
            'form_template_choices': [
                ('fobi/bootstrap3_extras/view_embed_form.html',
                 "Custom bootstrap3 embed form view template"),
            ],
            'success_page_template_choices': [
                ('fobi/bootstrap3_extras/embed_form_submitted.html',
                 "Custom bootstrap3 embed form entry submitted template"),
            ],
        },
    },
    'foundation5': {
        'feincms_integration': {
            'form_template_choices': [
                ('fobi/foundation5_extras/view_embed_form.html',
                 "Custom foundation5 embed form view template"),
            ],
            'success_page_template_choices': [
                ('fobi/foundation5_extras/embed_form_submitted.html',
                 "Custom foundation5 embed form entry submitted template"),
            ],
        },
    },
}
```

40.5.3 Usage

The `fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget` consists of the following fields:

- Form: The form to be used.
- Form template name: Template to be used to render the embed form.
- Hide form title: If checked, no form title would be shown.
- Form title: Overrides the standard form title.
- Submit button text: Overrides the default submit button text.
- Success page template name: Template to be used to render the embed form-sent event.
- Hide success page title: If checked, no form-sent title would be shown.
- Success page title: Overrides the form-sent title.
- Success page text: Overrides the form-sent text.

40.6 fobi.contrib.apps.mezzanine_integration

A django-fobi integration with Mezzanine.

40.6.1 Prerequisites

Tested with Mezzanine 3.1.10 and 4.2.3 only. Might work on earlier (or later) versions as well.

40.6.2 Installation

40.6.2.1 Versions

See the requirements files:

- [Mezzanine 4.2.3 requirements](#).

40.6.2.2 `your_project/settings.py`

See the [example settings file](#).

```
INSTALLED_APPS = list(INSTALLED_APPS)
INSTALLED_APPS += [
    # Standard mezzanine apps

    'fobi.contrib.apps.mezzanine_integration', # Fobi Mezzanine app
]
```

40.6.3 Information for developers

40.6.3.1 Template rendering

The form embed into Mezzanine page is rendered with use of two theme templates:

- `view_embed_form_entry_ajax_template`: Used for rendering the form.
- `embed_form_entry_submitted_ajax_template`: Used for rendering the form sent event.

40.6.3.2 Using custom templates for rendering the form

In the widget, you can specify a template which you want to be used for rendering the form or the form-sent event.

Example:

```
FOBI_MEZZANINE_INTEGRATION_FORM_TEMPLATE_CHOICES = (
    ("yourapp/custom_view_embed_form_v1.html",
     "Custom embed form view template #1"),
    ("yourapp/custom_view_embed_form_v2.html",
     "Custom embed form view template #2"),
)
```

Same goes for form-sent templates.

```
FOBI_MEZZANINE_INTEGRATION_SUCCESS_PAGE_TEMPLATE_CHOICES = (
    ("yourapp/custom_embed_form_submitted_v1.html",
     "Custom form-sent template #1"),
    ("yourapp/custom_embed_form_submitted_v2.html",
     "Custom form-sent template #2"),
)
```

Registering a template in the `FORM_TEMPLATE_CHOICES` makes it available for all the themes. If you rather want to use different custom templates for different themes, use the `FOBI_CUSTOM_THEME_DATA` as shown in the example below.

```
FOBI_CUSTOM_THEME_DATA = {
    'bootstrap3': {
        'mezzanine_integration': {
            'form_template_choices': [
                ('fobi/bootstrap3_extras/view_embed_form.html',
                 "Custom bootstrap3 embed form view template"),
            ],
            'success_page_template_choices': [
                ('fobi/bootstrap3_extras/embed_form_submitted.html',
                 "Custom bootstrap3 embed form entry submitted template"),
            ],
        },
    },
    'foundation5': {
        'mezzanine_integration': {
            'form_template_choices': [
                ('fobi/foundation5_extras/view_embed_form.html',
                 "Custom foundation5 embed form view template"),
            ],
            'success_page_template_choices': [
                ('fobi/foundation5_extras/embed_form_submitted.html',
                 "Custom foundation5 embed form entry submitted template"),
            ],
        },
    },
}
```

40.6.4 Usage

40.6.4.1 The fobi page model

The `fobi.contrib.apps.mezzanine_integration.models.FobiFormPage` consists of the following fields:

- Form: The form to be used.
- Form template name: Template to be used to render the embed form.
- Hide form title: If checked, no form title would be shown.
- Form title: Overrides the standard form title.
- Submit button text: Overrides the default submit button text.
- Success page template name: Template to be used to render the embed form-sent event.
- Hide success page title: If checked, no form-sent title would be shown.
- Success page title: Overrides the form-sent title.
- Success page text: Overrides the form-sent text.

40.6.4.2 Steps described

1. If you use the mezzanine [example](#) project, to start go to the <http://localhost:8003/fobi/> URL and create a form.

2. Then go to <http://localhost:8003/admin/pages/page/> and add a new *Fobi form* page.
3. Choose the form and optionally - override the form settings and then save the page.
4. See the page in the front-end.

40.7 fobi.contrib.plugins.form_elements.content.content_image

A Fobi Image form element plugin.

40.7.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_image` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'easy_thumbnails',  
    'fobi.contrib.plugins.form_elements.content.content_image',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_image.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_IMAGE_` to the desired variable name from the above mentioned defaults module.

40.8 fobi.contrib.plugins.form_elements.content.content_image_url

A Fobi ImageURL form element plugin.

40.8.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_image_url` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.content.content_image_url',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_image_url.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_IMAGE_URL_` to the desired variable name from the above mentioned `defaults` module.

40.9 fobi.contrib.plugins.form_elements.content.content_text

A Fobi Text form element plugin.

40.9.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.content.content_text',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_text.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_TEXT_` to the desired variable name from the above mentioned `defaults` module.

By default the content of the text field is stripped using either the awesome [bleach](#) library or if `bleach` is not installed just Django's own `strip_tags` function. To configure the strip (bleach only) behaviour, two settings are introduced:

```
- ALLOWED_TAGS:  
- ALLOWED_ATTRIBUTES:
```

The default values are:

```
ALLOWED_TAGS = [  
    'a',  
    'abbr',  
    'acronym',  
    'b',  
    'blockquote',  
    'code',  
    'em',
```

(continues on next page)

(continued from previous page)

```

        'i',
        'li',
        'ol',
        'strong',
        'ul',
    ]

    ALLOWED_ATTRIBUTES = {
        'a': ['href', 'title'],
        'abbr': ['title'],
        'acronym': ['title'],
    }

```

40.10 fobi.contrib.plugins.form_elements.content.content_richtext

A Fobi Rich text form element plugin based on [CKEditor](#) and [django-ckeditor](#).

40.10.1 Installation

- (1) Install `django-ckeditor`.

```
pip install django-ckeditor
```

- (2) Add `ckeditor` to `INSTALLED_APPS` in `settings.py`.

```

INSTALLED_APPS = (
    ...
    'ckeditor',
    ...
)

```

- (3) Add `fobi.contrib.plugins.form_elements.content.content_richtext` to `INSTALLED_APPS` in `settings.py`.

```

INSTALLED_APPS = (
    ...
    'fobi.contrib.plugins.form_elements.content.content_richtext',
    ...
)

```

- (4) Add `fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget` to `INSTALLED_APPS` in `settings.py` (if you're using `bootstrap3` theme). If you're using another theme, add correspondent widget specific to the active theme.

```

INSTALLED_APPS = (
    ...
    'fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_
↪bootstrap3_widget',
    ...
)

```

- (5) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (6) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.10.2 Controlling HTML tags and attributes

- (1) Install `bleach`.

```
pip install bleach
```

- (2) Specify `FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_TAGS`, `FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_ATTRIBUTES` and `FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_STYLES` in `settings.py`. The default values come from `bleach`:

```
FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_TAGS = [
    'a',
    'abbr',
    'acronym',
    'b',
    'blockquote',
    'code',
    'em',
    'i',
    'li',
    'ol',
    'strong',
    'ul',
]

FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_ATTRIBUTES = {
    'a': ['href', 'title'],
    'abbr': ['title'],
    'acronym': ['title'],
}

FOBI_PLUGIN_CONTENT_RICHTEXT_ALLOWED_STYLES = []
```

40.11 fobi.contrib.plugins.form_elements.content.content_video

A Fobi Video form element plugin.

40.11.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.content.content_video` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.content.content_video',
    # ...
)
```


- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Additionally, for the fine tuning, see the `fobi.contrib.plugins.form_elements.content.content_video.defaults` module. If necessary, override the settings by prepending `FOBI_PLUGIN_CONTENT_VIDEO_` to the desired variable name from the above mentioned defaults module.

40.12 fobi.contrib.plugins.form_elements.fields.boolean

A Fobi Boolean form field plugin. Makes use of the `django.forms.fields.BooleanField`.

40.12.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.boolean',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.13 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple

A Fobi Select Multiple form field plugin. Makes use of the `django.forms.fields.MultipleChoiceField` and `django.forms.widgets.CheckboxSelectMultiple`.

40.13.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default, the submitted form value of *checkbox_select_multiple* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

- “val”: *value* (example: “alpha”).
- “repr” (default): *label* (example: “Alpha”).
- “mix”: *value (label)* (example: “Alpha (alpha)”).

Simply set the `FOBI_FORM_ELEMENT_CHECKBOX_SELECT_MULTIPLE_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.13.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<ul id="id_NAME_OF_THE_ELEMENT">
  <li>
    <label for="id_NAME_OF_THE_ELEMENT_0">
      <input class="form-control" id="id_NAME_OF_THE_ELEMENT_0" name="NAME_OF_THE_
↪ELEMENT" type="checkbox" value="1" /> 1
    </label>
  </li>
  <li>
    <label for="id_NAME_OF_THE_ELEMENT_1">
      <input class="form-control" id="id_NAME_OF_THE_ELEMENT_1" name="NAME_OF_THE_
↪ELEMENT" type="checkbox" value="2" /> 2
    </label>
  </li>
  <li>
    <label for="id_NAME_OF_THE_ELEMENT_2">
      <input class="form-control" id="id_NAME_OF_THE_ELEMENT_2" name="NAME_OF_THE_
↪ELEMENT" type="checkbox" value="alpha" /> Alpha
```

(continues on next page)

(continued from previous page)

```

    </label>
</li>
<li>
    <label for="id_NAME_OF_THE_ELEMENT_3">
        <input class="form-control" id="id_NAME_OF_THE_ELEMENT_3" name="NAME_OF_THE_
↪ELEMENT" type="checkbox" value="beta" /> Beta
    </label>
</li>
<li>
    <label for="id_NAME_OF_THE_ELEMENT_4">
        <input class="form-control" id="id_NAME_OF_THE_ELEMENT_4" name="NAME_OF_THE_
↪ELEMENT" type="checkbox" value="omega" /> omega
    </label>
</li>
</ul>

```

40.14 fobi.contrib.plugins.form_elements.fields.date

A Fobi Date form field plugin. Makes use of the `django.forms.fields.DateField` and `django.forms.widgets.DateInput`.

40.14.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.date` to the `INSTALLED_APPS` in your `settings.py`.

```

INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.date',
    # ...
)

```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.15 fobi.contrib.plugins.form_elements.fields.date_drop_down

A Fobi Birthday form field plugin. Makes use of the `django.forms.fields.DateField` and `django.forms.extras.widgets.SelectDateWidget`.

40.15.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.date_drop_down` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.date_drop_down',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.16 fobi.contrib.plugins.form_elements.fields.datetime

A Fobi DateTime form field plugin. Makes use of the `django.forms.fields.DateTimeField` and `django.forms.widgets.DateTimeInput`.

40.16.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.datetime` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.datetime',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.17 fobi.contrib.plugins.form_elements.fields.decimal

A Fobi Decimal form field plugin. Makes use of the `django.forms.fields.DecimalField` and `django.forms.widgets.NumberInput` (falling back to `django.forms.widgets.TextInput` for older Django versions).

40.17.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.decimal` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.decimal',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.18 fobi.contrib.plugins.form_elements.fields.email

A Fobi Email form field plugin. Makes use of the `django.forms.fields.EmailField` and `django.forms.widgets.TextInput`.

40.18.1 Installation

(1) Add `fobi.contrib.plugins.form_elements.fields.email` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.email',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.19 fobi.contrib.plugins.form_elements.fields.file

A Fobi File form field plugin. Makes use of the `django.forms.fields.FileField` and `django.forms.widgets.ClearableFileInput`.

40.19.1 Installation

(1) Add `fobi.contrib.plugins.form_elements.fields.file` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.file',
```

(continues on next page)

(continued from previous page)

```
# ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default uploaded files are stored in the “fobi_plugins/file” directory of the media root. If you want to change the directory location, set the `FOBI_PLUGIN_FIELDS_FILE_FILES_UPLOAD_DIR` value to the desired (relative) path.
- (5) You may optionally restrict uploaded files extensions by specifying the `allowed_extensions` field in the plugin.

40.20 fobi.contrib.plugins.form_elements.fields.float

A Fobi Integer form field plugin. Makes use of the `django.forms.fields.FloatField` and `django.forms.widgets.NumberInput` (falling back to `django.forms.widgets.TextInput` for older Django versions).

40.20.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.float` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.float',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.21 fobi.contrib.plugins.form_elements.fields.hidden

A Fobi Hidden form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.HiddenInput`.

40.21.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.hidden` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.hidden',  
    # ...  
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.22 fobi.contrib.plugins.form_elements.fields.input

A generic input form field plugin. Makes use of the `django.forms.fields.Field` and `django.forms.widgets.Input`. Comes with a lot of options you likely won't use every day.

The full list of supported HTML properties is listed below:

- autocomplete
- autofocus
- disabled
- list
- max
- min
- multiple
- pattern
- placeholder
- readonly
- step
- type

See w3schools.com for further explanations.

40.22.1 Installation

(1) Add `fobi.contrib.plugins.form_elements.fields.input` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.input',  
    # ...  
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.23 fobi.contrib.plugins.form_elements.fields.integer

A Fobi Integer form field plugin. Makes use of the `django.forms.fields.IntegerField` and `django.forms.widgets.NumberInput` (falling back to `django.forms.widgets.TextInput` for older Django versions).

40.23.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.integer` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.integer',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.24 fobi.contrib.plugins.form_elements.fields.ip_address

A Fobi Text form field plugin. Makes use of the `django.forms.fields.GenericIPAddressField` and `django.forms.widgets.TextInput`.

40.24.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.ip_address` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.ip_address',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.25 fobi.contrib.plugins.form_elements.fields.null_boolean

A Fobi NullBoolean form field plugin. Makes use of the `django.forms.fields.NullBooleanField` and `django.forms.widgets.NullBooleanSelect`.

40.25.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.null_boolean` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.null_boolean',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FobiRestrictPluginAccess` is set to `True`.

40.26 fobi.contrib.plugins.form_elements.fields.password

A Fobi Password form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.PasswordInput`.

40.26.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.password` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.password',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FobiRestrictPluginAccess` is set to `True`.

40.27 fobi.contrib.plugins.form_elements.fields.radio

A Fobi Radio form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.RadioSelect`.

40.27.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.radio` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.radio',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default, the submitted form value of *radio* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[  
    ('alpha', 'Alpha'),  
    ('beta', 'Beta'),  
    ('gamma', 'Gamma'),  
]
```

```
- "val": `value` (example: "alpha").  
- "repr" (default): `label` (example: "Alpha").  
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_RADIO_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.27.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1  
2  
alpha, Alpha  
beta, Beta  
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">  
  <option value="1">1</option>  
  <option value="2">2</option>  
  <option value="alpha">Alpha</option>
```

(continues on next page)

(continued from previous page)

```
<option value="beta">Beta</option>
<option value="omega">omega</option>
</select>
```

40.28 fobi.contrib.plugins.form_elements.fields.range_select

A Fobi RangeSelect form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.Select`.

40.28.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.range_select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.range_select',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Ranges are specified within the given min/max values. The default values are:

```
- INITIAL: 50
- INITIAL_MAX_VALUE: 100
- INITIAL_MIN_VALUE: 0
- MIN_VALUE: 0
- MAX_VALUE: 100
- STEP: 1
```

However, you can override each of them in the settings of your project by prefixing correspondent names with `FOBI_FORM_ELEMENT_RANGE_SELECT_`:

```
- FOBI_FORM_ELEMENT_RANGE_SELECT_INITIAL
- FOBI_FORM_ELEMENT_RANGE_SELECT_INITIAL_MAX_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_INITIAL_MIN_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_MIN_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_MAX_VALUE
- FOBI_FORM_ELEMENT_RANGE_SELECT_STEP
```

- (5) By default, the submitted form value of `range_select` elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_RANGE_SELECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.29 fobi.contrib.plugins.form_elements.fields.regex

A Fobi Text form field plugin. Makes use of the `django.forms.fields.RegexField` and `django.forms.widgets.TextInput`.

40.29.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.regex` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.regex',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.30 fobi.contrib.plugins.form_elements.fields.select

A Fobi Select form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.Select`.

40.30.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select',

```

(continues on next page)

(continued from previous page)

```
# ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

(4) By default, the submitted form value of *select* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.30.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="alpha">Alpha</option>
  <option value="beta">Beta</option>
  <option value="omega">omega</option>
</select>
```

40.31 fobi.contrib.plugins.form_elements.fields.select_model_object

A Fobi Select Model Object form field plugin. Makes use of the `django.forms.models.ModelChoiceField` and `django.forms.widgets.Select`.

40.31.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_model_object` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.select_model_object',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_model_object.defaults.IGNORED_MODELS`. If necessary, override it in your *settings* as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MODEL_OBJECT_IGNORED_MODELS = [  
    'auth.User',  
    'auth.Group',  
]
```

- (5) By default, the submitted form value of *select_model_object* elements is `app_label.model_name.object_pk.object_repr`. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.  
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).  
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MODEL_OBJECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.32 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object

A Fobi Select MPTT Model Object form field plugin. Makes use of the `mptt.fields.TreeNodeChoiceField` and `django.forms.widgets.Select`.

40.32.1 Installation

40.32.1.1 Install *django-mptt*

Taken from *django-mptt* [Getting started](#).

1. Download `django-mptt` using `pip` by running:

```
pip install django-mptt
```

2. Add `mptt` to the `INSTALLED_APPS` in your `settings.py`.

40.32.1.2 Install `select_mptt_model_object` plugin

- (1) Add `mptt` and `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'mptt',
    'fobi.contrib.plugins.form_elements.fields.select_mptt_model_object',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.defaults.IGNORED_MODELS`. If necessary, override it in your *settings* as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MPTT_MODEL_OBJECT_IGNORED_MODELS = [
    'auth.User',
    'auth.Group',
]
```

- (5) By default, the submitted form value of `select_mptt_model_object` elements is `app_label.model_name.object_pk.object_repr`. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MPTT_MODEL_OBJECT_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.33 `fobi.contrib.plugins.form_elements.fields.select_multiple`

A Fobi Select Multiple form field plugin. Makes use of the `django.forms.fields.MultipleChoiceField` and `django.forms.widgets.SelectMultiple`.

40.33.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select_multiple',
    # ...
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

(4) By default, the submitted form value of *select_multiple* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.33.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">
  <option value="1">1</option>
  <option value="2">2</option>
  <option value="alpha">Alpha</option>
  <option value="beta">Beta</option>
  <option value="omega">omega</option>
</select>
```


40.34 fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects

A Fobi Select Multiple Model Objects form field plugin. Makes use of the `django.forms.models.ModelMultipleChoiceField` and `django.forms.widgets.SelectMultiple`.

40.34.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults.IGNORED_MODELS`. If necessary, override it in your *settings* as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MODEL_OBJECTS_IGNORED_MODELS = [
    'auth.User',
    'auth.Group',
]
```

- (5) By default, the submitted form value of *select_multiple_model_objects* elements is *app_label.model_name.object_pk.object_repr*. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MODEL_OBJECTS_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.35 fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects

A Fobi Select Multiple MPTT Model Objects form field plugin. Makes use of the `mptt.forms.TreeNodeMultipleChoiceField` and `django.forms.widgets.SelectMultiple`.

40.35.1 Installation

40.35.1.1 Install *django-mptt*

Taken from *django-mptt* [Getting started](#).

1. Download `django-mptt` using `pip` by running:

```
pip install django-mptt
```

2. Add `mptt` to the `INSTALLED_APPS` in your `settings.py`.

40.35.1.2 Install `select_multiple_mptt_model_objects` plugin

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'mptt',  
    'fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects'  
    ↪ ,  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Make sure to take a look at `fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults.IGNORED_MODELS`. If necessary, override it in your *settings* as shown in the example below:

```
FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MPTT_MODEL_OBJECTS_IGNORED_MODELS = [  
    'auth.User',  
    'auth.Group',  
]
```

- (5) By default, the submitted form value of `select_multiple_mptt_model_objects` elements is `app_label.model_name.object_pk.object_repr`. However, that part of the behaviour has been made configurable. You can choose between the following options:

```
- "val": `app_label.model_name.object_pk.object_repr`.  
- "repr": `object_repr` (uses the ``__unicode__`` method of the model).  
- "mix" (default): `app_label.model_name.object_pk.object_repr`.
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_MPTT_MODEL_OBJECTS_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.36 `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max`

A Fobi Select Multiple form field plugin with max choices. Makes use of the `django.forms.widgets.SelectMultiple`.

40.36.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.select_multiple_with_max` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.select_multiple_with_max',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) By default, the submitted form value of *select_multiple_with_max* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SELECT_MULTIPLE_WITH_MAX_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.36.2 Usage

You should be entering a single choice per line. Choice might consist of just a single value or value/label pair. If you enter an integer in the ‘max_choices’ field, the user can choose only <max_choices> or less choices.

For example:

```
1
2
alpha, Alpha
beta, Beta
omega
```

The following HTML would be made of:

```
<select id="id_NAME_OF_THE_ELEMENT" name="NAME_OF_THE_ELEMENT">
  <option value="1">1</option>
  <option value="2">2</option>
```

(continues on next page)

(continued from previous page)

```
<option value="alpha">Alpha</option>
<option value="beta">Beta</option>
<option value="omega">omega</option>
</select>
```

40.37 fobi.contrib.plugins.form_elements.fields.slider

A Fobi Percentage form field plugin. Makes use of the `django.forms.fields.ChoiceField` and `django.forms.widgets.Select`.

40.37.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.slider` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.slider',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Ranges are specified within the given min/max values. The default values are:

```
- INITIAL: 50
- INITIAL_MAX_VALUE: 100
- INITIAL_MIN_VALUE: 0
- MIN_VALUE: 0
- MAX_VALUE: 100
- STEP: 1
```

However, you can override each of them in the settings of your project by prefixing correspondent names with `FOBI_FORM_ELEMENT_SLIDER_`:

```
- FOBI_FORM_ELEMENT_SLIDER_INITIAL
- FOBI_FORM_ELEMENT_SLIDER_INITIAL_MAX_VALUE
- FOBI_FORM_ELEMENT_SLIDER_INITIAL_MIN_VALUE
- FOBI_FORM_ELEMENT_SLIDER_MIN_VALUE
- FOBI_FORM_ELEMENT_SLIDER_MAX_VALUE
- FOBI_FORM_ELEMENT_SLIDER_STEP
```

- (5) By default, the submitted form value of *slider* elements is label (human readable representation of the value chosen). However, that part of the behaviour has been made configurable. You can choose between the following options:

Consider the following list of (value, label) choices (the first element in the tuple is value, the second element is label):

```
[
    ('alpha', 'Alpha'),
    ('beta', 'Beta'),
    ('gamma', 'Gamma'),
]
```

```
- "val": `value` (example: "alpha").
- "repr" (default): `label` (example: "Alpha").
- "mix": `value (label)` (example: "Alpha (alpha)").
```

Simply set the `FOBI_FORM_ELEMENT_SLIDER_SUBMIT_VALUE_AS` assign one of the following values: “val”, “repr” or “mix” to get the desired behaviour.

40.38 fobi.contrib.plugins.form_elements.fields.text

A Fobi Text form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.TextInput`.

40.38.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.text` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.text',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.39 fobi.contrib.plugins.form_elements.fields.textarea

A Fobi Textarea form field plugin. Makes use of the `django.forms.fields.CharField` and `django.forms.widgets.Textarea`.

40.39.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.textarea` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.fields.textarea',

```

(continues on next page)

(continued from previous page)

```
# ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.40 fobi.contrib.plugins.form_elements.fields.time

A Fobi `DateTime` form field plugin. Makes use of the `django.forms.fields.TimeField` and `django.forms.widgets.TextInput`.

40.40.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.time` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.time',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.41 fobi.contrib.plugins.form_elements.fields.url

A Fobi `URL` form field plugin. Makes use of the `django.forms.fields.URLField` and `django.forms.widgets.URLInput` falling back to `django.forms.widgets.TextInput` for older Django versions.

40.41.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.fields.url` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.fields.url',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.42 fobi.contrib.plugins.form_elements.security.captcha

A **CAPTCHA** form field plugin. Makes use of the [django-simple-captcha](#).

40.42.1 Prerequisites

You will need `libfreetype6`, otherwise `django-simple-captcha` won't work.

```
sudo apt-get install libfreetype6-dev
```

40.42.2 Installation

40.42.2.1 Install *django-simple-captcha*

Taken from [django-simple-captcha installation instructions](#).

- (1) Download `django-simple-captcha` using `pip` by running:

```
pip install django-simple-captcha
```

- (2) Add `captcha` to the `INSTALLED_APPS` in your `settings.py`.
- (3) Run `python manage.py migrate`.
- (4) Add an entry to your `urls.py`:

```
urlpatterns += [
    url(r'^captcha/', include('captcha.urls')),
]
```

40.42.2.2 Install *fobi* Captcha plugin

- (1) Add `fobi.contrib.plugins.form_elements.security.captcha` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_elements.security.captcha',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

- (4) In order to have nicer text input widget, add the following line to your settings (if you're using bootstrap3 theme):

```
CAPTCHA_TEXT_FIELD_TEMPLATE = 'bootstrap3/captcha/text_field.html'
```

For foundation5 theme add the following line:

```
CAPTCHA_TEXT_FIELD_TEMPLATE = 'foundation5/captcha/text_field.html'
```

40.42.3 Troubleshooting and usage limitations

40.42.3.1 In combination with other captcha solutions

At the moment, you can't use both CAPTCHA (`fobi.contrib.plugins.form_elements.security.captcha`) and ReCAPTCHA (`fobi.contrib.plugins.form_elements.security.recaptcha`) plugins alongside due to app name collision of the `django-simple-captcha` and `django-recaptcha` packages.

40.42.4 Usage

Note, that unlike most of the other form element plugins, default value for the `required` attribute is `True`, which makes the Captcha obligatory. Although you could still set it to `False`, it does not make much sense to do so.

40.43 `fobi.contrib.plugins.form_elements.security.honeypot`

A [Honeypot](#) form field plugin. Just another anti-spam technique.

40.43.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.security.honeypot` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.security.honeypot',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.44 `fobi.contrib.plugins.form_elements.security.recaptcha`

A [ReCAPTCHA](#) form field plugin. Makes use of the `django-recaptcha`.

40.44.1 Installation

40.44.1.1 Install *django-recaptcha*

- (1) Download `django-recaptcha` using `pip` by running:

```
pip install django-recaptcha
```

- (2) Add `captcha` to the `INSTALLED_APPS` in your `settings.py`.
- (3) Run `python manage.py migrate`.

40.44.1.2 Install *fobi* ReCAPTCHA plugin

- (1) Add `fobi.contrib.plugins.form_elements.security.recaptcha` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.security.recaptcha',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Specify the following ReCAPTCHA credentials in your settings:

```
- ``RECAPTCHA_PUBLIC_KEY``  
- ``RECAPTCHA_PRIVATE_KEY``
```

For testing purposes obtain the keys [here](#):

40.44.2 Troubleshooting and usage limitations

40.44.2.1 In combination with other captcha solutions

At the moment, you can't use both `CAPTCHA` (`fobi.contrib.plugins.form_elements.security.captcha`) and `ReCAPTCHA` (`fobi.contrib.plugins.form_elements.security.recaptcha`) plugins alongside due to app name collision of the `django-simple-captcha` and `django-recaptcha` packages.

If you happen to see errors like “Input error: k: Format of site key was invalid”, make sure to have defined (and filled in properly) the `RECAPTCHA_PUBLIC_KEY` and `RECAPTCHA_PRIVATE_KEY` in your `settings.py`. See the [following](#) thread for more information.

40.44.3 Usage

Note, that unlike most of the other form element plugins, default value for the `required` attribute is `True`, which makes the ReCaptcha obligatory. Although you could still set it to `False`, it does not make much sense to do so.

40.45 fobi.contrib.plugins.form_elements.test.dummy

A Fobi Dummy form element plugin. Created for testing purposes.

40.45.1 Installation

- (1) Add `fobi.contrib.plugins.form_elements.test.dummy` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_elements.test.dummy',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.46 fobi.contrib.plugins.form_handlers.db_store

A Fobi Database Store form-/wizard- handler plugin. Saves submitted form data into the `SavedFormDataEntry`/`SavedFormWizardDataEntry` models.

40.46.1 Dependencies

The `xlwt` package is required (optional) for XLS export. If not present, export format falls back to CSV.

40.46.2 Installation

- (1) Add `fobi.contrib.plugins.form_handlers.db_store` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.plugins.form_handlers.db_store',  
    # ...  
)
```

- (2) In the terminal type:

```
./manage.py migrate  
  
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.
- (4) Add `db_store` form handler plugin URLs to the `urls.py` of your project.

```
urlpatterns = [
    # DB Store plugin URLs
    url(r'^fobi/plugins/form-handlers/db-store/',
        include('fobi.contrib.plugins.form_handlers.db_store.urls')),
]
```

For form wizards do:

```
urlpatterns = [
    # DB Store plugin URLs
    url(r'^fobi/plugins/form-wizard-handlers/db-store/',
        include('fobi.contrib.plugins.form_handlers.db_store.urls.'
                'form_wizard_handlers')),
]
```

40.47 fobi.contrib.plugins.form_handlers.http_repost

A Fobi HTTP Repost form handler plugin. Submits the form data as is to the given endpoint.

40.47.1 Installation

- (1) Add `fobi.contrib.plugins.form_handlers.http_repost` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_handlers.http_repost',
    # ...
)
```

- (2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

- (3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.48 fobi.contrib.plugins.form_handlers.mail

A Fobi Mail form handler plugin. Submits the form data by email to the specified email address.

40.48.1 Installation

- (1) Add `fobi.contrib.plugins.form_handlers.mail` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.plugins.form_handlers.mail',

```

(continues on next page)

(continued from previous page)

```
# ...  
)
```

(2) In the terminal type:

```
./manage.py fobi_sync_plugins
```

(3) Assign appropriate permissions to the target users/groups to be using the plugin if `FOBI_RESTRICT_PLUGIN_ACCESS` is set to `True`.

40.49 mailchimp_importer

A `django-fobi` integration with MailChimp.

This plugin makes it possible to import a form from a MailChimp list. A typical list URL would be [this](#). In the listing you would see list names and *Stats* at the right corner. If you click on it you would see the *Settings* link. Follow it and scroll to the bottom for the unique id for your list. Now, if you have been successfully authenticated to the MailChimp API using your `API_KEY`, you could call the `lists.merge_vars` method for getting the form. `API_KEY` could be obtained from the MailChimp in the [Account API](#).

For additional information on MailChimp import see the following [article](#).

40.49.1 Prerequisites

Python wrapper for the Mailchimp:

```
pip install mailchimp
```

If you are using Django 1.8 or greater, you would need `django-formtools` package as well:

```
pip install django-formtools
```

40.49.2 Installation

40.49.2.1 your_project/settings.py

```
INSTALLED_APPS = list(INSTALLED_APPS)  
INSTALLED_APPS += [  
    'fobi.contrib.plugins.form_importers.mailchimp_importer',  
]
```

40.49.3 How it works

Assuming that you have configured the `mailchimp_importer` plugin properly and have the Django running locally on port 8000, accessing the following URL would bring you to the MailChimp form import wizard.

- <http://localhost:8000/en/fobi/forms/importer/mailchimp/>

On the first step you would be asked to provide your `API_KEY`, which is used to authenticate to the MailChimp in order to fetch your list- and form- information. The key isn't stored/saved/remembered. Next time you want to import a form from the same account, you would have to provide it again.

40.49.4 Development status

This part of code is alpha, which means it experimental and needs improvements.

See the [TODOS](#) for the full list of planned-, pending- in-development- or to-be-implemented features.

If you want to improve it or did make it working, please, make a pull request.

40.50 fobi.contrib.themes.bootstrap3

A django-fobi Bootstrap 3 theme. Based on the ??? template.

40.50.1 Installation

- (1) Add `fobi.contrib.themes.bootstrap3` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (
    # ...
    'fobi.contrib.themes.bootstrap3',
    # ...
)
```

- (2) Specify `bootstrap3` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'bootstrap3'
```

40.51 fobi.contrib.themes.djangocms_admin_style_theme

A django-fobi theme in a style of `djangocms-admin-style` admin. Relies on `djangocms-admin-style` package and some jQuery UI only.

jQuery UI “Smoothness” theme comes from [here](#).

40.51.1 Installation

40.51.1.1 Install *djangocms-admin-style*

See the original [installation instructions](#).

- (1) Install the `djangocms-admin-style` package.

```
pip install djangocms-admin-style
```

- (2) Add `djangocms_admin_style` to your `INSTALLED_APPS` just before `django.contrib.admin`.

40.51.1.2 Install *fobi.contrib.themes.djangocms_admin_style_theme* theme

- (1) Add `fobi.contrib.themes.djangocms_admin_style_theme` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.themes.djangocms_admin_style_theme',  
    # ...  
)
```

- (2) Specify `djangocms_admin_style_theme` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'djangocms_admin_style_theme'
```

40.52 fobi.contrib.themes.foundation5

A django-fobi Foundation 5 theme. Based on the ??? template, but entire JS and CSS are taken from Foundation 5 version 5.4.0. The [following](#) icon set was used.

40.52.1 Installation

- (1) Add `fobi.contrib.themes.foundation5` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.themes.foundation5',  
    # ...  
)
```

- (2) Specify `foundation5` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'foundation5'
```

40.53 fobi.contrib.themes.simple

A django-fobi theme in a style of Django admin. Relies on Django-admin and some jQuery UI only. jQuery UI “Django” theme comes from [here](#).

40.53.1 Installation

- (1) Add `fobi.contrib.themes.simple` to the `INSTALLED_APPS` in your `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'fobi.contrib.themes.simple',  
    # ...  
)
```

- (2) Specify `simple` as a default theme in your `settings.py`:

```
FOBI_DEFAULT_THEME = 'simple'
```

- [1.1] Dashboard
- [1.2] Create a form
- [1.3] Edit form - form elements tab active, no elements yet
- [1.4] Edit form - form elements tab active, add a form element menu
- [1.5] Edit form - add a form element (URL plugin)
- [1.6] Edit form - form elements tab active, with form elements
- [1.7] Edit form - form handlers tab active, no handlers yet
- [1.8] Edit form - form handlers tab active, add form handler menu
- [1.9] Edit form - add a form handler (Mail plugin)
- [1.10] Edit form - form handlers tab active, with form handlers
- [1.11] Edit form - form properties tab active
- [1.12] View form
- [1.13] View form - form submitted (thanks page)
- [1.14] Edit form - add a form element (Video plugin)
- [1.15] Edit form - add a form element (Boolean plugin)
- [1.16] Edit form
- [1.17] View form
- [2.1] Edit form - form elements tab active, with form elements
- [2.2] Edit form - form elements tab active, add a form element menu
- [2.3] Edit form - add a form element (Hidden plugin)
- [2.4] Edit form - form handlers tab active, with form handlers
- [2.5] Edit form - form properties tab active
- [2.6] View form

Python Module Index

f

| | |
|---|--|
| fobi, 438 | fobi.contrib.apps.drf_integration.form_elements.content, 166 |
| fobi.admin, 373 | fobi.contrib.apps.drf_integration.form_elements.content, 166 |
| fobi.app, 377 | fobi.contrib.apps.drf_integration.form_elements.content, 167 |
| fobi.apps, 378 | fobi.contrib.apps.drf_integration.form_elements.content, 166 |
| fobi.base, 378 | fobi.contrib.apps.drf_integration.form_elements.content, 167 |
| fobi.conf, 397 | fobi.contrib.apps.drf_integration.form_elements.content, 166 |
| fobi.constants, 397 | fobi.contrib.apps.drf_integration.form_elements.content, 167 |
| fobi.context_processors, 397 | fobi.contrib.apps.drf_integration.form_elements.content, 168 |
| fobi.contrib, 354 | fobi.contrib.apps.drf_integration.form_elements.content, 167 |
| fobi.contrib.apps, 220 | fobi.contrib.apps.drf_integration.form_elements.content, 168 |
| fobi.contrib.apps.djangocms_integration, 165 | fobi.contrib.apps.drf_integration.form_elements.content, 167 |
| fobi.contrib.apps.djangocms_integration.apps, 164 | fobi.contrib.apps.drf_integration.form_elements.content, 168 |
| fobi.contrib.apps.djangocms_integration.conf, 164 | fobi.contrib.apps.drf_integration.form_elements.content, 167 |
| fobi.contrib.apps.djangocms_integration.defaults, 165 | fobi.contrib.apps.drf_integration.form_elements.content, 168 |
| fobi.contrib.apps.djangocms_integration.helpers, 165 | fobi.contrib.apps.drf_integration.form_elements.content, 168 |
| fobi.contrib.apps.djangocms_integration.settings, 165 | fobi.contrib.apps.drf_integration.form_elements.content, 170 |
| fobi.contrib.apps.drf_integration, 216 | fobi.contrib.apps.drf_integration.form_elements.content, 168 |
| fobi.contrib.apps.drf_integration.apps, 210 | fobi.contrib.apps.drf_integration.form_elements.content, 169 |
| fobi.contrib.apps.drf_integration.base, 210 | fobi.contrib.apps.drf_integration.form_elements.content, 169 |
| fobi.contrib.apps.drf_integration.dynamic, 212 | fobi.contrib.apps.drf_integration.form_elements.content, 171 |
| fobi.contrib.apps.drf_integration.fields, 212 | fobi.contrib.apps.drf_integration.form_elements.content, 170 |
| fobi.contrib.apps.drf_integration.form_elements, 207 | fobi.contrib.apps.drf_integration.form_elements.content, 170 |
| fobi.contrib.apps.drf_integration.form_elements.Content, 172 | fobi.contrib.apps.drf_integration.form_elements.content, 170 |
| fobi.contrib.apps.drf_integration.form_elements.Content.content_image, 166 | fobi.contrib.apps.drf_integration.form_elements.content, 172 |
| fobi.contrib.apps.drf_integration.form_elements.Content.content_image.apps, 166 | fobi.contrib.apps.drf_integration.form_elements.content, 172 |

| | |
|---|-------------------------------------|
| 203 | 209 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappextrappsintegration.form_handlers.ma | |
| 202 | 209 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappextrbasintegration.helpers, | |
| 203 | 214 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappextrfobintegrationhelpers, | |
| 203 | 214 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappextrfobintegrationserializers, | |
| 204 | 214 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappextrfobintegrationurls, | |
| 203 | 215 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappextrfobintegrationutils, | |
| 204 | 215 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappextrfobintegrationview_form_elements, | |
| 204 | 215 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappsmfeincms_integration, | |
| 205 | 218 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappsmfeapps_integration.apps, | |
| 204 | 216 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappsmfebasms_integration.conf, | |
| 205 | 216 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappsmfebomisintegrationdefaults, | |
| 205 | 216 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappslfeincms_integration.helpers, | |
| 207 | 216 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappslfeappsms_integration.settings, | |
| 205 | 216 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappslfbasms_integration.widgets, | |
| 206 | 217 |
| fobi.contrib.apps.drf_integration.form_elements.fobidappslmeobanintegrationform_elements, | |
| 206 | 220 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.apps.mezzanine_integration.apps, | |
| 210 | 218 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.apps.mezzanine_integration.conf, | |
| 208 | 218 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.apps.mezzanine_integration.defaults, | |
| 207 | 219 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.apps.mezzanine_integration.helpers, | |
| 207 | 219 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.apps.mezzanine_integration.handlers, | |
| 207 | 219 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.plugins, | 335 |
| 209 | fobi.contrib.plugins.form_elements, |
| | 313 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.plugins, | |
| 208 | 237 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.plugins, | |
| 208 | 223 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.plugins, | |
| 208 | 220 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.plugins, | |
| 210 | 220 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.plugins, | |
| 209 | 221 |
| fobi.contrib.apps.drf_integration.form_handlers.fobidcontrib.plugins, | |

| | |
|--|--|
| 221 | 232 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 221 | 232 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 222 | 233 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 223 | 233 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 223 | 233 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 226 | 234 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 223 | 237 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 223 | 234 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 224 | 234 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 224 | 235 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 224 | 235 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 225 | 235 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 225 | 236 |
| fobi.contrib.plugins.form_elements.content | fobi.contrib.plugins.form_elements.content |
| 228 | 237 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 226 | 305 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 226 | 238 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 227 | 237 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 228 | 237 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 228 | 237 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 231 | 238 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 228 | 241 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 229 | 238 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 229 | 239 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 231 | 239 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 231 | 240 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 234 | 240 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |
| 231 | 241 |
| fobi.contrib.plugins.form_elements.fields | fobi.contrib.plugins.form_elements.fields |

| | |
|--|---|
| 241 | 251 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.email, |
| 243 | 253 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.email.app, |
| 241 | 251 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.email.base, |
| 242 | 252 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.email.fobi, |
| 242 | 252 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.email.fobi, |
| 243 | 253 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file, |
| 243 | 256 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file.app, |
| 245 | 253 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file.base, |
| 243 | 253 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file.conf, |
| 244 | 254 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file.default, |
| 244 | 255 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file.fobi, |
| 245 | 255 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file.fobi, |
| 247 | 256 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.file.set, |
| 245 | 256 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.float, |
| 246 | 258 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.float.app, |
| 246 | 256 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.float.base, |
| 247 | 257 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.float.fobi, |
| 247 | 257 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.float.fobi, |
| 249 | 258 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.hidden, |
| 247 | 260 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.hidden.app, |
| 248 | 258 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.hidden.base, |
| 248 | 258 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.hidden.fobi, |
| 249 | 259 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.hidden.fobi, |
| 251 | 259 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.input, |
| 249 | 262 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.input.app, |
| 250 | 260 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.input.base, |
| 250 | 260 |
| fobi.contrib.plugins.form_elements.fields.date | fobi.contrib.plugins.form_elements.fields.input.conf, |

| | |
|--|---|
| 295 | 304 |
| fobi.contrib.plugins.form_elements.fields.fobidenthelp | fobi.contrib.plugins.form_elements.fields.url.forms |
| 296 | 305 |
| fobi.contrib.plugins.form_elements.fields.fobidentsebtphggs | fobi.contrib.plugins.form_elements.security, |
| 296 | 311 |
| fobi.contrib.plugins.form_elements.fields.fobidentwidgpt | fobi.contrib.plugins.form_elements.security.captcha |
| 296 | 307 |
| fobi.contrib.plugins.form_elements.fields.foblugontrib | fobi.contrib.plugins.form_elements.security.captcha |
| 298 | 305 |
| fobi.contrib.plugins.form_elements.fields.foblugonapps | fobi.contrib.plugins.form_elements.security.captcha |
| 296 | 306 |
| fobi.contrib.plugins.form_elements.fields.foblugobaseb | fobi.contrib.plugins.form_elements.security.captcha |
| 297 | 306 |
| fobi.contrib.plugins.form_elements.fields.foblugofbifbfrmgenform | fobi.contrib.plugins.form_elements.security.captcha |
| 297 | 307 |
| fobi.contrib.plugins.form_elements.fields.foblugoftrib | fobi.contrib.plugins.form_elements.security.honeypot |
| 298 | 310 |
| fobi.contrib.plugins.form_elements.fields.fobextontrib | fobi.contrib.plugins.form_elements.security.honeypot |
| 300 | 307 |
| fobi.contrib.plugins.form_elements.fields.fobextonapps | fobi.contrib.plugins.form_elements.security.honeypot |
| 298 | 307 |
| fobi.contrib.plugins.form_elements.fields.fobextobaseb | fobi.contrib.plugins.form_elements.security.honeypot |
| 298 | 308 |
| fobi.contrib.plugins.form_elements.fields.fobextofbifbfrmgenform | fobi.contrib.plugins.form_elements.security.honeypot |
| 299 | 308 |
| fobi.contrib.plugins.form_elements.fields.fobextoftrib | fobi.contrib.plugins.form_elements.security.honeypot |
| 299 | 308 |
| fobi.contrib.plugins.form_elements.fields.fobextantrib | fobi.contrib.plugins.form_elements.security.honeypot |
| 301 | 309 |
| fobi.contrib.plugins.form_elements.fields.fobextantribapps | fobi.contrib.plugins.form_elements.security.honeypot |
| 300 | 309 |
| fobi.contrib.plugins.form_elements.fields.fobextantribbaseb | fobi.contrib.plugins.form_elements.security.honeypot |
| 300 | 309 |
| fobi.contrib.plugins.form_elements.fields.fobextantribfoblugontrib | fobi.contrib.plugins.form_elements.security.recaptcha |
| 300 | 311 |
| fobi.contrib.plugins.form_elements.fields.fobextantribfoblugontrib | fobi.contrib.plugins.form_elements.security.recaptcha |
| 301 | 310 |
| fobi.contrib.plugins.form_elements.fields.fobimeontrib | fobi.contrib.plugins.form_elements.security.recaptcha |
| 303 | 310 |
| fobi.contrib.plugins.form_elements.fields.fobimeonapps | fobi.contrib.plugins.form_elements.security.recaptcha |
| 301 | 310 |
| fobi.contrib.plugins.form_elements.fields.fobimeobaseb | fobi.contrib.plugins.form_elements.security.recaptcha |
| 302 | 311 |
| fobi.contrib.plugins.form_elements.fields.fobimeofbifbfrmgenform | fobi.contrib.plugins.form_elements.test, |
| 302 | 313 |
| fobi.contrib.plugins.form_elements.fields.fobimeoftrib | fobi.contrib.plugins.form_elements.test.dummy, |
| 303 | 313 |
| fobi.contrib.plugins.form_elements.fields.fobirilcontrib | fobi.contrib.plugins.form_elements.test.dummy.apps, |
| 305 | 311 |
| fobi.contrib.plugins.form_elements.fields.fobirilapps | fobi.contrib.plugins.form_elements.test.dummy.base, |
| 303 | 312 |
| fobi.contrib.plugins.form_elements.fields.fobirilbaseb | fobi.contrib.plugins.form_elements.test.dummy.fobi, |
| 304 | 312 |
| fobi.contrib.plugins.form_elements.fields.fobirilcontribfoblugontrib | fobi.contrib.plugins.form_elements.test.dummy.widg |

312 fobi.contrib.plugins.form_handlers.mail.base,
 fobi.contrib.plugins.form_handlers, 332 326
 fobi.contrib.plugins.form_handlers.db_store, 327
 322 327
 fobi.contrib.plugins.form_handlers.db_store.admin, 328
 314 328
 fobi.contrib.plugins.form_handlers.db_store.apps, 328
 315 328
 fobi.contrib.plugins.form_handlers.db_store.base, 328
 315 328
 fobi.contrib.plugins.form_handlers.db_store.callbacks, 329
 316 329
 fobi.contrib.plugins.form_handlers.db_store.conf, 330
 317 330
 fobi.contrib.plugins.form_handlers.db_store.defaults, 330
 317 330
 fobi.contrib.plugins.form_handlers.db_store.fields, 331
 317 331
 fobi.contrib.plugins.form_handlers.db_store.helpers, 331
 318 331
 fobi.contrib.plugins.form_handlers.db_store.importers, 331
 314 331
 fobi.contrib.plugins.form_handlers.db_store.mailchimp, 335
 313 335
 fobi.contrib.plugins.form_handlers.db_store.migrations, 335
 313 335
 fobi.contrib.plugins.form_handlers.db_store.models, 332
 319 332
 fobi.contrib.plugins.form_handlers.db_store.settings, 333
 321 333
 fobi.contrib.plugins.form_handlers.db_store.urls, 333
 314 333
 fobi.contrib.plugins.form_handlers.db_store.urls.form_handlers, 334
 314 334
 fobi.contrib.plugins.form_handlers.db_store.urls.form_wizard_handlers, 354
 314 354
 fobi.contrib.plugins.form_handlers.db_store.views, 341
 321 341
 fobi.contrib.plugins.form_handlers.db_store.widgets, 339
 322 339
 fobi.contrib.plugins.form_handlers.http_post, 339
 326 339
 fobi.contrib.plugins.form_handlers.http_post.apps, 339
 322 339
 fobi.contrib.plugins.form_handlers.http_post.base, 336
 323 336
 fobi.contrib.plugins.form_handlers.http_post.fobi_form_handlers, 335
 324 335
 fobi.contrib.plugins.form_handlers.http_post.forms, 335
 325 335
 fobi.contrib.plugins.form_handlers.mail, fobi.contrib.themes.bootstrap3.widgets.form_element, 336
 332 336
 fobi.contrib.plugins.form_handlers.mail.apps, contrib.themes.bootstrap3.widgets.form_element, 336
 326 336

```

fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3_widget.fobi_themes,
336 fobi.contrib.themes.foundation5.widgets,
fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget,
337 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps,
336 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements,
337 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget,
338 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps,
337 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.fobi_form_elements,
337 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget,
338 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps,
338 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements,
338 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget,
339 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.apps,
338 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3_widget.fobi_form_elements,
339 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes,
345 fobi.contrib.themes.foundation5.widgets.form_elements,
fobi.contrib.themes.djangocms_admin_style_theme.apps,
343 fobi.contrib.themes.foundation5.widgets.form_handlers,
fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes,
343 fobi.contrib.themes.foundation5.widgets.form_handlers,
fobi.contrib.themes.djangocms_admin_style_theme.widgets,
343 fobi.contrib.themes.foundation5.widgets.form_handlers,
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements,
342 fobi.contrib.themes.foundation5.widgets.form_handlers,
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_widget,
342 fobi.contrib.themes.simple, 354
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_widget,
341 fobi.contrib.themes.simple.fobi_themes,
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_admin_widget,
342 fobi.contrib.themes.simple.widgets, 352
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers,
343 fobi.contrib.themes.simple.widgets, 352
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers,
343 fobi.contrib.themes.simple.widgets, 352
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers,
342 fobi.contrib.themes.simple.widgets, 351
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers,
342 fobi.contrib.themes.simple.widgets, 351
fobi.contrib.themes.foundation5, 351 fobi.contrib.themes.simple.widgets.form_handlers,
fobi.contrib.themes.foundation5.apps, 352
349 fobi.contrib.themes.simple.widgets.form_handlers.django,
fobi.contrib.themes.foundation5.fobi_themes, 352

```

- fobi.contrib.themes.simple.widgets.form_handlers, 352
- fobi.contrib.themes.simple.widgets.form_handlers.markdown_widget, 352
- fobi.data_structures, 398
- fobi.decorators, 399
- fobi.defaults, 400
- fobi.discover, 400
- fobi.dynamic, 401
- fobi.exceptions, 401
- fobi.form_importers, 403
- fobi.form_utils, 404
- fobi.forms, 404
- fobi.helpers, 409
- fobi.integration, 356
- fobi.integration.helpers, 354
- fobi.integration.processors, 355
- fobi.management, 357
- fobi.management.commands, 357
- fobi.management.commands.fobi_find_broken_entries, 356
- fobi.management.commands.fobi_sync_plugins, 356
- fobi.management.commands.fobi_update_plugin_data, 356
- fobi.migrations, 359
- fobi.migrations.0001_initial, 357
- fobi.migrations.0002_auto_20150912_1744, 357
- fobi.migrations.0003_auto_20160517_1005, 357
- fobi.migrations.0004_auto_20160906_1513, 357
- fobi.migrations.0005_auto_20160908_1457, 358
- fobi.migrations.0006_auto_20160911_1549, 358
- fobi.migrations.0007_auto_20160926_1652, 358
- fobi.migrations.0008_formwizardhandlerentry, 358
- fobi.migrations.0009_formwizardentry_wizard_type, 358
- fobi.migrations.0010_formwizardhandler, 358
- fobi.migrations.0011_formentry_title, 359
- fobi.migrations.0012_auto_20161109_1550, 359
- fobi.migrations.0013_formwizardentry_show_all_navigation_buttons, 359
- fobi.migrations.0014_auto_20170516_1413, 359
- fobi.models, 414
- fobi.reusable, 360
- fobi.reusable.markdown_widget, 360
- fobi.reusable.markdown_widget.widgets, 360
- fobi.settings, 429
- fobi.templatetags, 362
- fobi.templatetags.fobi_tags, 360
- fobi.templatetags.future_compat, 362
- fobi.tests, 367
- fobi.tests.base, 362
- fobi.tests.constants, 363
- fobi.tests.core, 363
- fobi.tests.data, 363
- fobi.tests.helpers, 363
- fobi.tests.test_browser_build_dynamic_forms, 364
- fobi.tests.test_core, 365
- fobi.tests.test_drf_integration, 365
- fobi.tests.test_dynamic_forms, 366
- fobi.tests.test_feincms_integration, 366
- fobi.tests.test_form_importers_mailchimp, 366
- fobi.tests.test_sortable_dict, 367
- fobi.urls, 367
- fobi.urls.edit, 367
- fobi.urls.view, 367
- fobi.utils, 429
- fobi.validators, 432
- fobi.views, 432
- fobi.widgets, 438
- fobi.wizard, 373
- fobi.wizard.views, 373
- fobi.wizard.views.dynamic, 367
- fobi.wizard.views.views, 372

A

abstract (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget.Meta attribute), 217
 abstract (fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataEntry.Meta attribute), 319
 abstract (fobi.models.AbstractFormWizardPluginEntry.Meta attribute), 419
 abstract (fobi.models.AbstractPluginEntry.Meta attribute), 419
 abstract (fobi.models.AbstractPluginModel.Meta attribute), 415
 abstract (fobi.models.BaseAbstractPluginEntry.Meta attribute), 420
 AbstractFormWizardPluginEntry (class in fobi.models), 418
 AbstractFormWizardPluginEntry.Meta (class in fobi.models), 418
 AbstractPluginEntry (class in fobi.models), 419
 AbstractPluginEntry.Meta (class in fobi.models), 419
 AbstractPluginModel (class in fobi.models), 414
 AbstractPluginModel.Meta (class in fobi.models), 415
 AbstractSavedFormDataEntry (class in fobi.contrib.plugins.form_handlers.db_store.models), 319
 AbstractSavedFormDataEntry.Meta (class in fobi.contrib.plugins.form_handlers.db_store.models), 319
 action (fobi.models.FormEntry attribute), 421
 actions (fobi.admin.FormElementAdmin attribute), 374
 actions (fobi.admin.FormHandlerAdmin attribute), 375
 actions (fobi.admin.FormWizardHandlerAdmin attribute), 377
 actions (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntryAdmin attribute), 314
 actions (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntryAdmin attribute), 314
 add_arguments() (fobi.management.commands.fobi_sync_plugins.Command method), 356
 add_form_element_entry() (in module fobi.views), 432
 add_form_element_entry_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 339
 add_form_element_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.Bootstrap3Theme attribute), 343
 add_form_element_entry_ajax_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 349
 add_form_element_entry_ajax_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 353
 add_form_element_entry_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 339
 add_form_element_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.Bootstrap3Theme attribute), 343
 add_form_element_entry_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 349
 add_form_element_entry_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 353
 add_form_handler_entry() (in module fobi.views), 433
 add_form_handler_entry_ajax_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 339
 add_form_handler_entry_ajax_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.Bootstrap3Theme attribute), 343

[attribute](#)), 343
[add_form_handler_entry_ajax_template](#) ([fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#) attribute), 349
[add_form_handler_entry_ajax_template](#) ([fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#) attribute), 353
[add_form_handler_entry_template](#) ([fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#) attribute), 339
[add_form_handler_entry_template](#) ([fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme](#) attribute), 343
[add_form_handler_entry_template](#) ([fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#) attribute), 349
[add_form_handler_entry_template](#) ([fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#) attribute), 353
[add_form_template](#) ([fobi.base.BasePlugin](#) attribute), 379
[add_form_wizard_form_entry\(\)](#) (in module [fobi.views](#)), 433
[add_form_wizard_handler_entry\(\)](#) (in module [fobi.views](#)), 433
[add_form_wizard_handler_entry_ajax_template](#) ([fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#) attribute), 339
[add_form_wizard_handler_entry_template](#) ([fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#) attribute), 339
[admin_change_url\(\)](#) (in module [fobi.helpers](#)), 409
[all_permissions_required\(\)](#) (in module [fobi.decorators](#)), 400
[allow_multiple](#) ([fobi.base.FormHandlerPlugin](#) attribute), 387
[allow_multiple](#) ([fobi.base.FormWizardHandlerPlugin](#) attribute), 389
[allow_multiple](#) ([fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin](#) attribute), 315
[allow_multiple](#) ([fobi.contrib.plugins.form_handlers.db_store.base.DBStoreWizardHandlerPlugin](#) attribute), 316
[allow_multiple](#) ([fobi.contrib.plugins.form_handlers.db_store.fobi_themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme](#) attribute), 317
[allow_multiple](#) ([fobi.contrib.plugins.form_handlers.db_store.fobi_themes.simple.fobi_themes.SimpleTheme](#) attribute), 318
[any_permission_required\(\)](#) (in module [fobi.decorators](#)), 400
[app_config\(\)](#) (in module [fobi.app](#)), 377
[app_label](#) ([fobi.admin.BasePluginModelAdmin.Meta](#) attribute), 373
[app_label](#) ([fobi.admin.FormElementEntryAdmin.Meta](#) attribute), 374
[app_label](#) ([fobi.admin.FormEntryAdmin.Meta](#) attribute), 375
[app_label](#) ([fobi.admin.FormFieldsetEntryAdmin.Meta](#) attribute), 375
[app_label](#) ([fobi.admin.FormHandlerEntryAdmin.Meta](#) attribute), 376
[app_label](#) ([fobi.admin.FormWizardEntryAdmin.Meta](#) attribute), 376
[app_label](#) ([fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget](#) attribute), 217
[app_label](#) ([fobi.contrib.plugins.form_handlers.db_store.admin.SavedForm](#) attribute), 314
[app_label](#) ([fobi.contrib.plugins.form_handlers.db_store.admin.SavedForm](#) attribute), 314
[app_name\(\)](#) (in module [fobi.app](#)), 377
[as_delete_and_delete_links_to_field\(\)](#) (in module [fobi.utils](#)), 429
[as_text\(\)](#) ([fobi.form_utils.ErrorDict](#) method), 404
[as_text\(\)](#) ([fobi.form_utils.ErrorList](#) method), 404
[as_view\(\)](#) ([fobi.wizard.views.dynamic.DynamicWizardView](#) class method), 367
[assemble_form_class\(\)](#) (in module [fobi.dynamic](#)), 401
[assemble_form_field_widget_class\(\)](#) (in module [fobi.base](#)), 378
[assemble_form_wizard_class\(\)](#) (in module [fobi.dynamic](#)), 401
[assemble_serializer_class\(\)](#) (in module [fobi.contrib.apps.drf_integration.dynamic](#)), 212
[auto_discover\(\)](#) (in module [fobi.discover](#)), 400
[AutoFormDbStore](#) (class in [fobi.contrib.plugins.form_handlers.db_store.callbacks](#)), 316
[AutoFormMail](#) (class in [fobi.contrib.plugins.form_handlers.mail.callbacks](#)), 327

B

[base_base.DBStoreHandlerPlugin](#) (in module [fobi.admin](#)), 373
[base_base.DBStoreWizardHandlerPlugin](#) ([fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme](#) attribute), 344
[base_edit_template](#) ([fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#) attribute), 353
[base_fields](#) ([fobi.contrib.plugins.form_elements.content.content_image](#) attribute), 222
[base_fields](#) ([fobi.contrib.plugins.form_elements.content.content_image](#) attribute), 225
[base_fields](#) ([fobi.contrib.plugins.form_elements.content.content_mark](#) attribute), 228
[base_fields](#) ([fobi.contrib.plugins.form_elements.content.content_richt](#) attribute), 231

base_fields (fobi.contrib.plugins.form_elements.content_text.forms.TextInputForm attribute), 233

base_fields (fobi.contrib.plugins.form_elements.content_image.forms.ImageField attribute), 233

base_fields (fobi.contrib.plugins.form_elements.content_video.forms.VideoForm attribute), 236

base_fields (fobi.contrib.plugins.form_elements.fields.boolean.forms.BooleanField attribute), 238

base_fields (fobi.contrib.plugins.form_elements.fields.checkbox.forms.BooleanField attribute), 241

base_fields (fobi.contrib.plugins.form_elements.fields.date.forms.DateField attribute), 243

base_fields (fobi.contrib.plugins.form_elements.fields.date_drop_down.forms.DateField attribute), 245

base_fields (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeField attribute), 247

base_fields (fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalField attribute), 249

base_fields (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeField attribute), 251

base_fields (fobi.contrib.plugins.form_elements.fields.email.forms.EmailForm attribute), 253

base_fields (fobi.contrib.plugins.form_elements.fields.file.forms.FileForm attribute), 256

base_fields (fobi.contrib.plugins.form_elements.fields.float.forms.FloatField attribute), 258

base_fields (fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenForm attribute), 259

base_fields (fobi.contrib.plugins.form_elements.fields.input.forms.InputForm attribute), 261

base_fields (fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerField attribute), 263

base_fields (fobi.contrib.plugins.form_elements.fields.ip_address.forms.IPAddressForm attribute), 264

base_fields (fobi.contrib.plugins.form_elements.fields.multiple_choice.forms.MultipleChoiceForm attribute), 266

base_fields (fobi.contrib.plugins.form_elements.fields.password.forms.PasswordForm attribute), 268

base_fields (fobi.contrib.plugins.form_elements.fields.radio.forms.RadioButton attribute), 270

base_fields (fobi.contrib.plugins.form_elements.fields.range_select.forms.RangeSelectForm attribute), 273

base_fields (fobi.contrib.plugins.form_elements.fields.regex.forms.RegexForm attribute), 275

base_fields (fobi.contrib.plugins.form_elements.fields.select.forms.SelectForm attribute), 277

base_fields (fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectForm attribute), 280

base_fields (fobi.contrib.plugins.form_elements.fields.select_model.forms.SelectModelForm attribute), 282

base_fields (fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleForm attribute), 285

base_fields (fobi.contrib.plugins.form_elements.fields.select_multiple_and_one.forms.SelectMultipleAndOneForm attribute), 287

base_fields (fobi.contrib.plugins.form_elements.fields.select_multiple_and_one_model_object.forms.SelectMultipleAndOneModelObjectForm attribute), 290

base_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme (class in fobi.contrib.themes.bootstrap3.fobi_themes), 339
fobi.contrib.plugins.form_elements.fields.boolean.forms),
base_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme (class in fobi.contrib.themes.djangocms_admin_style_theme), 343
BooleanSelectPlugin (class in fobi.contrib.plugins.form_elements.fields.boolean.forms),
base_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme (class in fobi.contrib.themes.foundation5.fobi_themes), 349
172
base_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme (class in fobi.contrib.themes.simple.fobi_themes), 353
fobi.contrib.apps.drf_integration.form_elements.fields.boolean.forms),
base_view_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme (class in fobi.contrib.themes.djangocms_admin_style_theme), 343
fobi.contrib.plugins.form_elements.fields.boolean.base),
base_view_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme (class in fobi.contrib.themes.simple.fobi_themes), 353
BooleanSelectPlugin (class in fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_elements),
BaseAbstractPluginEntry (class in fobi.models), 237
419 Bootstrap3Theme (class in fobi.contrib.themes.bootstrap3.fobi_themes),
BaseAbstractPluginEntry.Meta (class in fobi.models), 420 339
BaseContentMarkdownPluginWidget (class in fobi.contrib.plugins.form_elements.content.content_markdown.widgets), 228
bulk_change_form_element_plugins () (in module fobi.admin), 374
BaseContentRichTextPluginWidget (class in fobi.contrib.plugins.form_elements.content.richtext.widgets), 231
bulk_change_form_handler_plugins () (in module fobi.admin), 374
BaseDataStorage (class in fobi.base), 378
bulk_change_plugins ()
BaseDatePluginWidget (class in fobi.contrib.plugins.form_elements.fields.date.widgets), 243
(fobi.admin.BasePluginModelAdmin method), 373
BaseDateTimePluginWidget (class in fobi.contrib.plugins.form_elements.fields.datetime.widgets), 247
BulkChangeFormElementPluginsForm (class in fobi.forms), 404
(class in fobi.forms), 404
BaseDbStorePluginWidget (class in fobi.contrib.plugins.form_handlers.db_store.widgets), 322
BulkChangeFormHandlerPluginsForm (class in fobi.forms), 404
(class in fobi.forms), 404
BaseDummyPluginWidget (class in fobi.contrib.plugins.form_elements.test.dummy.widgets), 312
BulkChangeFormWizardHandlerPluginsForm (class in fobi.forms), 405
(class in fobi.forms), 405
BaseException, 401
BulkChangeFormWizardHandlerPluginsForm.Meta (class in fobi.forms), 405
BaseFobiBrowserBuldDynamicFormsTest (class in fobi.tests.base), 362
BaseFormFieldPluginForm (class in fobi.base), 378
C
callback () (fobi.base.FormCallback method), 385
callback () (fobi.base.IntegrationFormCallback method), 394
callback () (fobi.contrib.plugins.form_handlers.db_store.callbacks.AutoFormCallback method), 316
callback () (fobi.contrib.plugins.form_handlers.mail.callbacks.AutoFormCallback method), 327
can_redirect (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 217
can_redirect (fobi.integration.processors.IntegrationProcessor attribute), 355
CaptchaInputForm (class in fobi.contrib.plugins.form_elements.security.captcha.forms),

307
 CaptchaInputPlugin (class in clean (fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm method), 305
 fobi.contrib.plugins.form_elements.security.captcha.base.CaptchaForm (fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotForm method), 308
 306
 CaptchaInputPlugin (class in clean_action (fobi.forms.FormEntryForm method), 406
 fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaForm (fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotForm method), 409
 306
 CheckboxSelectMultipleInputForm (class in clean_initial (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.ChrisForm method), 241
 fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.ChrisForm (fobi.contrib.plugins.form_elements.fields.date.forms.DateForm method), 243
 241
 CheckboxSelectMultipleInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple.ChrisForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 247
 173
 CheckboxSelectMultipleInputPlugin (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.ChrisForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 247
 174
 CheckboxSelectMultipleInputPlugin (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.ChrisForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 247
 174
 CheckboxSelectMultipleInputPlugin (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.ChrisForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 247
 239
 CheckboxSelectMultipleInputPlugin (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.ChrisForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 247
 240
 ClassProperty (class in fobi.base), 384
 classproperty (in module fobi.base), 384
 clean (fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 249
 method), 249
 clean (fobi.contrib.plugins.form_elements.fields.duration.forms.DurationForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 251
 method), 251
 clean (fobi.contrib.plugins.form_elements.fields.email.forms.EmailForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 253
 method), 253
 clean (fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 256
 method), 256
 clean (fobi.contrib.plugins.form_elements.fields.float.forms.FloatInputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 258
 method), 258
 clean (fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenInputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 260
 method), 260
 clean (fobi.contrib.plugins.form_elements.fields.input.forms.InputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 261
 method), 261
 clean (fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerInputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 263
 method), 263
 clean (fobi.contrib.plugins.form_elements.fields.ip_address.forms.IPAddressForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 264
 method), 264
 clean (fobi.contrib.plugins.form_elements.fields.password.forms.PasswordForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 268
 method), 268
 clean (fobi.contrib.plugins.form_elements.fields.range_select.forms.RangeSelectInputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 273
 method), 273
 clean (fobi.contrib.plugins.form_elements.fields.regex.forms.RegexForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 275
 method), 275
 clean (fobi.contrib.plugins.form_elements.fields.slider.forms.SliderForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 295
 method), 295
 clean (fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 298
 method), 298
 clean (fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeFieldForm method), 299
 method), 299

Config (class in fobi.contrib.plugins.form_elements.fields.date_time(class in fobi.contrib.plugins.form_elements.fields.time.apps),
 243 301
 Config (class in fobi.contrib.plugins.form_elements.fields.datetime(class in fobi.contrib.plugins.form_elements.fields.url.apps),
 245 303
 Config (class in fobi.contrib.plugins.form_elements.fields.decimal(class in fobi.contrib.plugins.form_elements.security.captcha.apps),
 247 305
 Config (class in fobi.contrib.plugins.form_elements.fields.duration(class in fobi.contrib.plugins.form_elements.security.honeypot.apps),
 249 307
 Config (class in fobi.contrib.plugins.form_elements.fields.email(class in fobi.contrib.plugins.form_elements.security.recaptcha.apps),
 251 310
 Config (class in fobi.contrib.plugins.form_elements.fields.file(class in fobi.contrib.plugins.form_elements.test.dummy.apps),
 253 311
 Config (class in fobi.contrib.plugins.form_elements.fields.float(class in fobi.contrib.plugins.form_handlers.db_store.apps),
 256 315
 Config (class in fobi.contrib.plugins.form_elements.fields.hidden(class in fobi.contrib.plugins.form_handlers.http_repost.apps),
 258 322
 Config (class in fobi.contrib.plugins.form_elements.fields.input(class in fobi.contrib.plugins.form_handlers.mail.apps),
 260 326
 Config (class in fobi.contrib.plugins.form_elements.fields.integer(class in fobi.contrib.plugins.form_importers.mailchimp_importer.apps),
 262 332
 Config (class in fobi.contrib.plugins.form_elements.fields.ip_address(class in fobi.contrib.themes.bootstrap3.apps),
 263 339
 Config (class in fobi.contrib.plugins.form_elements.fields.multiple_choice(class in fobi.contrib.themes.bootstrap3.widgets.form_elements.combobox.apps),
 265 335
 Config (class in fobi.contrib.plugins.form_elements.fields.password(class in fobi.contrib.themes.bootstrap3.widgets.form_elements.combobox.apps),
 266 336
 Config (class in fobi.contrib.plugins.form_elements.fields.radio(class in fobi.contrib.themes.bootstrap3.widgets.form_elements.django_bootstrap_form.apps),
 268 336
 Config (class in fobi.contrib.plugins.form_elements.fields.range_select(class in fobi.contrib.themes.bootstrap3.widgets.form_elements.django_bootstrap_form.apps),
 271 337
 Config (class in fobi.contrib.plugins.form_elements.fields.regex(class in fobi.contrib.themes.bootstrap3.widgets.form_elements.django_bootstrap_form.apps),
 273 338
 Config (class in fobi.contrib.plugins.form_elements.fields.select(class in fobi.contrib.themes.bootstrap3.widgets.form_elements.django_bootstrap_form.apps),
 275 338
 Config (class in fobi.contrib.plugins.form_elements.fields.select_in(class in fobi.contrib.themes.djangocms_admin_style_theme.apps),
 278 343
 Config (class in fobi.contrib.plugins.form_elements.fields.select_in(class in fobi.contrib.themes.djangocms_admin_style_theme.widgets.django_bootstrap_form.apps),
 280 341
 Config (class in fobi.contrib.plugins.form_elements.fields.select_in(class in fobi.contrib.themes.djangocms_admin_style_theme.widgets.django_bootstrap_form.apps),
 282 342
 Config (class in fobi.contrib.plugins.form_elements.fields.select_in(class in fobi.contrib.themes.djangocms_admin_style_theme.widgets.django_bootstrap_form.apps),
 285 349
 Config (class in fobi.contrib.plugins.form_elements.fields.select_in(class in fobi.contrib.themes.djangocms_admin_style_theme.widgets.django_bootstrap_form.apps),
 288 345
 Config (class in fobi.contrib.plugins.form_elements.fields.select_in(class in fobi.contrib.themes.djangocms_admin_style_theme.widgets.django_bootstrap_form.apps),
 290 346
 Config (class in fobi.contrib.plugins.form_elements.fields.slide(class in fobi.contrib.themes.foundation5.widgets.form_elements.django_bootstrap_form.apps),
 293 346
 Config (class in fobi.contrib.plugins.form_elements.fields.slug(class in fobi.contrib.themes.foundation5.widgets.form_elements.django_bootstrap_form.apps),
 296 347
 Config (class in fobi.contrib.plugins.form_elements.fields.text(class in fobi.contrib.themes.foundation5.widgets.form_handlers.django_bootstrap_form.apps),
 298 348
 Config (class in fobi.contrib.plugins.form_elements.fields.textarea(class in fobi.contrib.themes.simple.apps), 352
 300 Config (class in fobi.contrib.themes.simple.widgets.form_elements.contenteditable(class in fobi.contrib.themes.simple.widgets.form_elements.django_bootstrap_form.apps), 352

```

351 Config (class in fobi.contrib.themes.simple.widgets.form_elements.django_admin_style_theme), 213
352 ContentImageField (class in ContentRichTextForm (class in
fobi.contrib.apps.drf_integration.fields), 212 fobi.contrib.plugins.form_elements.content.content_richtext.forms),
ContentImageForm (class in 231
fobi.contrib.plugins.form_elements.content.content_image.forms), ContentTextPlugin (class in
222 fobi.contrib.apps.drf_integration.form_elements.content.content_
ContentImagePlugin (class in 169
fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base),
166 fobi.contrib.apps.drf_integration.form_elements.content.content_
ContentImagePlugin (class in 169
fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base),
166 fobi.contrib.plugins.form_elements.content.content_richtext.base),
ContentImagePlugin (class in 229
fobi.contrib.plugins.form_elements.content.content_richtext.base), ContentRichTextPlugin (class in
220 fobi.contrib.plugins.form_elements.content.content_richtext.fobi_
ContentImagePlugin (class in 229
fobi.contrib.plugins.form_elements.content.content_image.form_elements), ContentFormWidget (class in
221 fobi.contrib.themes.bootstrap3.widgets.form_elements.content_r
ContentImageURLForm (class in 336
fobi.contrib.plugins.form_elements.content.content_image.url.forms), ContentTextPluginWidget (class in
225 fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_
ContentImageURLPlugin (class in 342
fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base), ContentRichTextPluginWidget (class in
167 fobi.contrib.themes.foundation5.widgets.form_elements.content_r
ContentImageURLPlugin (class in 345
fobi.contrib.apps.drf_integration.form_elements.content.content_richtext.base),
167 fobi.contrib.apps.drf_integration.form_elements.content_richtext.fobi_
ContentImageURLPlugin (class in 351
fobi.contrib.plugins.form_elements.content.content_image.url.base), ContentImageField (class in
223 fobi.contrib.apps.drf_integration.fields), 213
ContentImageURLPlugin (class in ContentTextForm (class in
fobi.contrib.plugins.form_elements.content.content_image_fobi.contrib.plugins.form_elements.content.content_text.forms),
224 233
ContentMarkdownField (class in ContentTextPlugin (class in
fobi.contrib.apps.drf_integration.fields), 212 fobi.contrib.apps.drf_integration.form_elements.content.content_
ContentMarkdownForm (class in 170
fobi.contrib.plugins.form_elements.content.content_markdown.forms), ContentTextPlugin (class in
228 fobi.contrib.apps.drf_integration.form_elements.content.content_
ContentMarkdownPlugin (class in 170
fobi.contrib.apps.drf_integration.form_elements.content.content_text.base),
168 fobi.contrib.plugins.form_elements.content.content_text.base),
ContentMarkdownPlugin (class in 232
fobi.contrib.apps.drf_integration.form_elements.content.content_text.fobi_integr(class form_elements),
168 fobi.contrib.plugins.form_elements.content.content_text.fobi_form
ContentMarkdownPlugin (class in 233
fobi.contrib.plugins.form_elements.content.content_markdown.base), ContentImageField (class in
226 fobi.contrib.apps.drf_integration.fields), 213
ContentMarkdownPlugin (class in ContentVideoForm (class in
fobi.contrib.plugins.form_elements.content.content_markdown_fobi.contrib.plugins.form_elements.content.content_video.forms),
227 236
ContentMarkdownPluginWidget (class in ContentVideoPlugin (class in
fobi.contrib.themes.bootstrap3.widgets.form_elements.content_fobi.contrib.apps.drf_integration.widgets.fobi_form_elements),

```


171
ContentVideoPlugin (class in (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme
fobi.contrib.apps.drf_integration.form_elements.content.content_video.base), 349
171
ContentVideoPlugin (class in (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme
fobi.contrib.plugins.form_elements.content.content_video.base), 349
234
ContentVideoPlugin (class in (fobi.contrib.themes.simple.fobi_themes.SimpleTheme
fobi.contrib.plugins.form_elements.content.content_video.fobi_themes), 353
235
convert_to_markdown() (in module (fobi.contrib.plugins.form_handlers.db_store.models.AbstractSa
fobi.reusable.markdown_widget.helpers), created (fobi.contrib.plugins.form_handlers.db_store.models.AbstractSa
359
CookieWizardView (class in custom_actions() (fobi.base.FormHandlerPlugin
fobi.wizard.views.views), 373 method), 387
copy() (fobi.data_structures.SortableDict method), 398 custom_actions() (fobi.base.FormWizardHandlerPlugin
create_form_entry() (in module fobi.views), 433 method), 389
create_form_entry_ajax_template custom_actions() (fobi.contrib.plugins.form_handlers.db_store.base.
(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme method), 315
attribute), 339 custom_actions() (fobi.contrib.plugins.form_handlers.db_store.base.
create_form_entry_ajax_template method), 316
(fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme
attribute), 343 method), 317
create_form_entry_ajax_template custom_actions() (fobi.contrib.plugins.form_handlers.db_store.fobi_
(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme method), 318
attribute), 349
create_form_entry_ajax_template D
(fobi.contrib.themes.simple.fobi_themes.SimpleTheme dashboard() (in module fobi.views), 434
attribute), 353 dashboard_template
create_form_entry_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme
(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340
attribute), 339 dashboard_template
create_form_entry_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.
(fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme
attribute), 343 attribute), 343
create_form_entry_template dashboard_template
(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 349
attribute), 349 dashboard_template
create_form_entry_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme
(fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 353
attribute), 353
create_form_with_entries() (in module DataExporter (class in
fobi.tests.helpers), 363 fobi.contrib.plugins.form_handlers.db_store.helpers),
318
create_form_wizard_entry() (in module DateDropDownInputForm (class in
fobi.views), 434 fobi.contrib.plugins.form_elements.fields.date_drop_down.forms),
245
create_form_wizard_entry_ajax_template DateDropDownInputPlugin (class in
(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 339 fobi.contrib.plugins.form_elements.fields.date_drop_down.base),
244
create_form_wizard_entry_ajax_template DateDropDownInputPlugin (class in
(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 349 fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_
244
create_form_wizard_entry_ajax_template DateDropDownSelectPlugin (class in
(fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 353 fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_

176 `fobi.contrib.plugins.form_handlers.db_store.base),`
DateDropDownSelectPlugin (class in 315
`fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_down_fobi_integration (class elements), in`
176 `fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers)`
DateInputForm (class in 317
`fobi.contrib.plugins.form_elements.fields.date.forms),`
243 `DBStorePluginWidget (class in`
DateInputPlugin (class in 342
`fobi.contrib.plugins.form_elements.fields.date.base),`
242 `DBStorePluginWidget (class in`
DateInputPlugin (class in 348
`fobi.contrib.plugins.form_elements.fields.date.fobi_form_elements), inWidget (class in`
242 `fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_f`
DatePluginWidget (class in 352
`fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget_fobi_form_ele(class), in`
337 `fobi.contrib.plugins.form_handlers.db_store.base),`
DatePluginWidget (class in 316
`fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget_fobi_form_ele(class), in`
346 `fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers)`
DateSelectPlugin (class in 318
`fobi.contrib.apps.drf_integration.form_elements.fields.date.base),`
175 `DecimalInputForm (class in`
DateSelectPlugin (class in 249
`fobi.contrib.plugins.form_elements.fields.decimal.forms),`
175 `fobi.contrib.apps.drf_integration.form_elements.fields.date_fobi_integration form_element(class in`
`fobi.contrib.apps.drf_integration.form_elements.fields.decimal.base)`
DateTimeInputForm (class in 178
`fobi.contrib.plugins.form_elements.fields.datetime.forms),`
247 `DecimalInputPlugin (class in`
DateTimeInputPlugin (class in 178
`fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi`
246 `DecimalInputPlugin (class in`
DateTimeInputPlugin (class in 248
`fobi.contrib.plugins.form_elements.fields.decimal.base),`
246 `fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_elements), in`
`fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_eleme`
DateTimePluginWidget (class in 248
`fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget_fobi_plugins(form_elements),`
337 `attribute), 222`
DateTimePluginWidget (class in declared_fields (fobi.contrib.plugins.form_elements.content.content_
`fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget_fobi_form_elements),`
347 `declared_fields (fobi.contrib.plugins.form_elements.content.content_`
DateTimeSelectPlugin (class in attribute), 228
`fobi.contrib.apps.drf_integration.form_elements.fields.datetime.base), ds (fobi.contrib.plugins.form_elements.content.content_`
177 `attribute), 231`
DateTimeSelectPlugin (class in declared_fields (fobi.contrib.plugins.form_elements.content.content_
`fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_integration form_elements),`
177 `declared_fields (fobi.contrib.plugins.form_elements.content.content_`
`attribute), 236`
db_clean_up () (in module fobi.tests.helpers), 363
DBStoreHandlerPlugin (class in declared_fields (fobi.contrib.plugins.form_elements.fields.boolean.fobi_
`fobi.contrib.apps.drf_integration.form_handlers.db_store.base), attribute), 238`
207 `declared_fields (fobi.contrib.plugins.form_elements.fields.checkbox_`
DBStoreHandlerPlugin (class in attribute), 241
`fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration (fobi.contrib.plugins.form_elements.fields.date.forms`
207 `attribute), 243`
DBStoreHandlerPlugin (class in declared_fields (fobi.contrib.plugins.form_elements.fields.date_drop

[illegible]

[attribute](#)), 308
[delete_file\(\)](#) (in module [fobi.helpers](#)), 409
[delete_form_element_entry\(\)](#) (in module [fobi.views](#)), 434
[delete_form_entry\(\)](#) (in module [fobi.views](#)), 434
[delete_form_handler_entry\(\)](#) (in module [fobi.views](#)), 434
[delete_form_wizard_entry\(\)](#) (in module [fobi.views](#)), 434
[delete_form_wizard_form_entry\(\)](#) (in module [fobi.views](#)), 435
[delete_plugin_data\(\)](#) ([fobi.base.BasePlugin](#) method), 379
[delete_plugin_data\(\)](#) ([fobi.contrib.plugins.form_elements.content.content_image.FobiContentImagePlugin](#) method), 220
[delete_plugin_data\(\)](#) ([fobi.contrib.plugins.form_elements.content.content_image.FobiContentImagePlugin](#) method), 221
[dependencies](#) ([fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial.Migration](#) attribute), 313
[dependencies](#) ([fobi.contrib.plugins.form_handlers.db_store.migrations.0002_auto_20150912_1744.Migration](#) attribute), 313
[dependencies](#) ([fobi.migrations.0001_initial.Migration](#) attribute), 357
[dependencies](#) ([fobi.migrations.0002_auto_20150912_1744.Migration](#) attribute), 357
[dependencies](#) ([fobi.migrations.0003_auto_20160517_1005.Migration](#) attribute), 357
[dependencies](#) ([fobi.migrations.0004_auto_20160906_1513.Migration](#) attribute), 357
[dependencies](#) ([fobi.migrations.0005_auto_20160908_1457.Migration](#) attribute), 358
[dependencies](#) ([fobi.migrations.0006_auto_20160911_1549.Migration](#) attribute), 358
[dependencies](#) ([fobi.migrations.0007_auto_20160926_1652.Migration](#) attribute), 358
[dependencies](#) ([fobi.migrations.0008_formwizardhandlerentry.Migration](#) attribute), 358
[dependencies](#) ([fobi.migrations.0009_formwizardentry_wizard_type.Migration](#) attribute), 358
[dependencies](#) ([fobi.migrations.0010_formwizardhandlerentry.Migration](#) attribute), 358
[dependencies](#) ([fobi.migrations.0011_formentry_title.Migration](#) attribute), 359
[dependencies](#) ([fobi.migrations.0012_auto_20161109_1550.Migration](#) attribute), 359
[dependencies](#) ([fobi.migrations.0013_formwizardentry_show_all_on_navigation_buttons.Migration](#) attribute), 359
[dependencies](#) ([fobi.migrations.0014_auto_20170516_1413.Migration](#) attribute), 359
[description](#) ([fobi.base.BasePlugin](#) attribute), 379
[description](#) ([fobi.form_importers.BaseFormImporter](#) attribute), 403
[dispatch\(\)](#) ([fobi.wizard.views.dynamic.DynamicWizardView](#) method), 368
[DjangoCMSAdminStyleTheme](#) (class in [fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes](#)), 343
[do_http_repost\(\)](#) ([fobi.contrib.plugins.form_handlers.http_repost.ba](#) method), 323
[do_http_repost\(\)](#) ([fobi.contrib.plugins.form_handlers.http_repost.fobi](#) method), 324
[do_slugify\(\)](#) (in module [fobi.helpers](#)), 409
[DoesNotExist](#), 401
[done\(\)](#) ([fobi.contrib.plugins.form_importers.mailchimp_importer.views.M](#) method), 334
[done\(\)](#) ([fobi.views.FormWizardView](#) method), 437
[done_step_name](#) ([fobi.wizard.views.dynamic.DynamicNamedUrlWizar](#) attribute), 374
[DRFIntegrationFormElementPluginProcessor](#) (class in [fobi.contrib.plugins.form_handlers.drf_integration.base](#)), 210
[DummyPlugin](#) (class in [fobi.contrib.plugins.form_elements.test.dummy.base](#)), 312
[DummyPlugin](#) (class in [fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elemen](#)), 312
[DummyPluginWidget](#) (class in [fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bo](#)), 318
[DummyPluginWidget](#) (class in [fobi.contrib.themes.foundation5.widgets.form_elements.dummy_fo](#)), 347
[DurationInputForm](#) (class in [fobi.contrib.plugins.form_elements.fields.duration.forms](#)), 250
[DurationInputPlugin](#) (class in [fobi.contrib.apps.drf_integration.form_elements.fields.duration.bo](#)), 179
[DurationInputPlugin](#) (class in [fobi.contrib.apps.drf_integration.form_elements.fields.duration.fobi](#)), 179
[DurationInputPlugin](#) (class in [fobi.contrib.plugins.form_elements.fields.duration.base](#)), 250
[dynamic_values\(\)](#) (in module [fobi.context_processors](#)), 397
[DynamicCookieWizardView](#) (class in [fobi.wizard.views.dynamic](#)), 371

DynamicNamedUrlCookieWizardView (class in *fobi.wizard.views.dynamic*), 372

DynamicNamedUrlSessionWizardView (class in *fobi.wizard.views.dynamic*), 372

DynamicNamedUrlWizardView (class in *fobi.wizard.views.dynamic*), 371

DynamicSessionWizardView (class in *fobi.wizard.views.dynamic*), 371

DynamicWizardView (class in *fobi.wizard.views.dynamic*), 367

E

edit_form_element_entry() (in module *fobi.views*), 435

edit_form_element_entry_ajax_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 340

edit_form_element_entry_ajax_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 343

edit_form_element_entry_ajax_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_element_entry_ajax_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_element_entry_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 340

edit_form_element_entry_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 343

edit_form_element_entry_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_element_entry_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_element_entry_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 340

edit_form_element_entry_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 343

edit_form_element_entry_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_element_entry_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_entry() (in module *fobi.views*), 435

edit_form_entry_ajax_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 340

edit_form_entry_ajax_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 343

edit_form_entry_ajax_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_entry_ajax_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_entry_edit_option_html() (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* class method), 343

edit_form_entry_help_text_extra() (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* class method), 343

edit_form_entry_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 340

edit_form_entry_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 343

edit_form_entry_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_entry_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_handler_entry() (in module *fobi.views*), 435

edit_form_handler_entry_ajax_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 340

edit_form_handler_entry_ajax_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 344

edit_form_handler_entry_ajax_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_handler_entry_ajax_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_handler_entry_template (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 340

edit_form_handler_entry_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 344

edit_form_handler_entry_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_handler_entry_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_template (*fobi.base.BasePlugin* attribute), 379

edit_form_wizard_entry_ajax_template (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 349

edit_form_wizard_entry_ajax_template (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 353

edit_form_wizard_entry_template (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme* attribute), 343

| | |
|---|---|
| <code>attribute</code>), 340 | <code>(fobi.contrib.themes.djangoCMS_admin_style_theme.widgets.form</code> |
| <code>edit_form_wizard_entry_template</code> | <code>attribute</code>), 342 |
| <code>(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> | <code>entries_icon_class</code> |
| <code>attribute</code>), 349 | <code>(fobi.contrib.themes.foundation5.widgets.form_handlers.db_store</code> |
| <code>edit_form_wizard_entry_template</code> | <code>attribute</code>), 348 |
| <code>(fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> | <code>export_entries_icon_class</code> |
| <code>attribute</code>), 353 | <code>(fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_</code> |
| <code>edit_form_wizard_handler_entry()</code> (in mod- | <code>attribute</code>), 352 |
| <code>ule fobi.views</code>), 435 | <code>export_form_entry()</code> (in module <code>fobi.views</code>), 436 |
| <code>edit_form_wizard_handler_entry_ajax_template</code> | <code>export_saved_form_data_entries()</code> (in mod- |
| <code>(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> | <code>ule fobi.contrib.plugins.form_handlers.db_store.views</code>), |
| <code>attribute</code>), 340 | 321 |
| <code>edit_form_wizard_handler_entry_template</code> | <code>export_saved_form_wizard_data_entries()</code> |
| <code>(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> | <code>in module fobi.contrib.plugins.form_handlers.db_store.views</code>), |
| <code>attribute</code>), 340 | 322 |
| <code>email</code> (<code>fobi.helpers.StrippedUser</code> attribute), 413 | <code>export_to_csv()</code> (<code>fobi.contrib.plugins.form_handlers.db_store.helpers</code> |
| <code>EmailInputForm</code> (class in | <code>method</code>), 318 |
| <code>fobi.contrib.plugins.form_elements.fields.email.form</code>), | <code>export_to_json()</code> (<code>fobi.helpers.JSONDataExporter</code> |
| 253 | <code>method</code>), 412 |
| <code>EmailInputPlugin</code> (class in | <code>export_to_xls()</code> (<code>fobi.contrib.plugins.form_handlers.db_store.helpers</code> |
| <code>fobi.contrib.apps.drf_integration.form_elements.fields.email.base</code>), | <code>extra</code> (<code>fobi.admin.FormElementEntryInlineAdmin</code> at- |
| 180 | <code>tribute</code>), 374 |
| <code>EmailInputPlugin</code> (class in | <code>extra</code> (<code>fobi.admin.FormElementEntryInlineAdmin</code> at- |
| <code>fobi.contrib.apps.drf_integration.form_elements.fields.email.base</code>), | <code>tribute</code>), 376 |
| 181 | <code>tribute</code>), 376 |
| <code>EmailInputPlugin</code> (class in | <code>extra</code> (<code>fobi.admin.FormWizardFormEntryInlineAdmin</code> |
| <code>fobi.contrib.plugins.form_elements.fields.email.base</code>), | <code>attribute</code>), 377 |
| 252 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>EmailInputPlugin</code> (class in | <code>attribute</code>), 377 |
| <code>fobi.contrib.plugins.form_elements.fields.email.base</code>), | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| 252 | <code>attribute</code>), 377 |
| <code>embed_form_entry_submitted_ajax_template</code> | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>(fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 340 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>ensure_autodiscover()</code> (in module <code>fobi.base</code>), | <code>attribute</code>), 377 |
| 384 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>ensure_autodiscover()</code> (in module | <code>attribute</code>), 377 |
| <code>fobi.form_importers</code>), 403 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>ensure_unique_filename()</code> (in module | <code>attribute</code>), 377 |
| <code>fobi.helpers</code>), 409 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>entry_user</code> (<code>fobi.models.AbstractFormWizardPluginEntry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 419 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>entry_user</code> (<code>fobi.models.AbstractPluginEntry</code> at- | <code>attribute</code>), 377 |
| <code>tribute</code>), 419 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>entry_user</code> (<code>fobi.models.BaseAbstractPluginEntry</code> at- | <code>attribute</code>), 377 |
| <code>tribute</code>), 420 | <code>extra</code> (<code>fobi.admin.FormWizardHandlerEntryInlineAdmin</code> |
| <code>ErrorDict</code> (class in <code>fobi.form_utils</code>), 404 | <code>attribute</code>), 377 |
| <code>ErrorList</code> (class in <code>fobi.form_utils</code>), 404 | <code>attribute</code>), 377 |
| <code>export()</code> (<code>fobi.helpers.JSONDataExporter</code> method), | <code>attribute</code>), 377 |
| 412 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| <code>export_entries_icon_class</code> | <code>attribute</code>), 377 |
| <code>(fobi.contrib.plugins.form_handlers.db_store.widgets.BaseFormWizardHandlerPluginRegistry</code> | <code>attribute</code>), 377 |
| <code>attribute</code>), 322 | <code>attribute</code>), 377 |
| | |

(*fobi.base.IntegrationFormElementPluginRegistry* attribute), 407
 attribute), 395 fields (*fobi.forms.FormWizardHandlerEntryForm.Meta* attribute), 408
 fail_on_missing_plugin (*fobi.base.IntegrationFormHandlerPluginRegistry* fields_mapping (*fobi.contrib.plugins.form_importers.mailchimp_importer* attribute), 396 attribute), 332
 FeinCMSIntegrationTest (class in fields_mapping (*fobi.contrib.plugins.form_importers.mailchimp_importer* attribute), 366 attribute), 333
 field_properties_mapping fields_mapping (*fobi.form_importers.BaseFormImporter* attribute), 332 base_mailchimp_importer attribute), 332
 field_properties_mapping fieldsets (*fobi.admin.BasePluginModelAdmin* attribute), 373
 field_properties_mapping (*fobi.contrib.plugins.form_importers.mailchimp_importer* fields_mapping (*fobi.contrib.plugins.form_importers.mailchimp_importer* attribute), 333 attribute), 374
 field_properties_mapping fieldsets (*fobi.admin.FormEntryAdmin* attribute), 375
 field_properties_mapping (*fobi.form_importers.BaseFormImporter* fieldsets (*fobi.admin.FormFieldsetEntryAdmin* attribute), 403 attribute), 375
 field_type_prop_name (*fobi.contrib.plugins.form_importers.mailchimp_importer* fields_mapping (*fobi.contrib.plugins.form_importers.mailchimp_importer* attribute), 332 attribute), 376
 field_type_prop_name fieldsets (*fobi.admin.FormWizardEntryAdmin* attribute), 333 *fobi.form_importers* in *fobi.contrib.plugins.form_importers.mailchimp_importer* attribute), 333
 field_type_prop_name fieldsets (*fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormEntryAdmin* attribute), 403 attribute), 314
 fields (*fobi.admin.FormElementEntryInlineAdmin* attribute), 375 file_storage (*fobi.views.FormWizardView* attribute), 437
 fields (*fobi.admin.FormHandlerEntryInlineAdmin* attribute), 376 FileInputForm (class in *fobi.contrib.plugins.form_elements.fields.file.forms*), 256
 fields (*fobi.admin.FormWizardFormEntryInlineAdmin* attribute), 377 FileInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.file.base*), 181
 fields (*fobi.admin.FormWizardHandlerEntryInlineAdmin* attribute), 377 *fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer.Meta* (class in *fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_in*), 214
 fields (*fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer.Meta* attribute), 214 182
 fields (*fobi.forms.BulkChangeFormElementPluginsForm.Meta* attribute), 404 FileInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.file.base*), 253
 fields (*fobi.forms.BulkChangeFormHandlerPluginsForm.Meta* attribute), 404 253
 fields (*fobi.forms.BulkChangeFormWizardHandlerPluginsForm.Meta* attribute), 405 FileInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.file.fobi_form_elements*), 255
 fields (*fobi.forms.FormElementEntryForm.Meta* attribute), 405 filter_horizontal (*fobi.admin.BasePluginModelAdmin* attribute), 373
 fields (*fobi.forms.FormEntryForm.Meta* attribute), 405 373
 fields (*fobi.forms.FormFieldsetEntryForm.Meta* attribute), 406 finalize() (*fobi.contrib.apps.feincms_integration.widgets.FobiFormWizard* method), 217
 fields (*fobi.forms.FormHandlerEntryForm.Meta* attribute), 406 fire_form_callbacks() (in module *fobi.base*), 384
 fields (*fobi.forms.FormHandlerForm.Meta* attribute), 407 firstof() (in module *fobi.templatetags.future_compat*), 362
 fields (*fobi.forms.FormWizardEntryForm.Meta* attribute), 407 flatatt_inverse_quotes() (in module *fobi.helpers*), 410
 fields (*fobi.forms.FormWizardFormEntryForm.Meta* FloatInputForm (class in

[fobi.contrib.plugins.form_elements.fields.float.form](#) (module), 166
 258
 FloatInputPlugin (class in [fobi.contrib.apps.drf_integration.form_elements](#) (module), 166
[fobi.contrib.apps.drf_integration.form_elements.fields.float.base](#) (module), 166
 183
 FloatInputPlugin (class in [fobi.contrib.apps.drf_integration.form_elements](#) (module), 167
[fobi.contrib.apps.drf_integration.form_elements.fields.float.fobi_integration](#) (module), 166
 183
 FloatInputPlugin (class in [fobi.contrib.apps.drf_integration.form_elements](#) (module), 167
[fobi.contrib.plugins.form_elements.fields.float.base](#) (module), 167
 257
 FloatInputPlugin (class in [fobi.contrib.apps.drf_integration.form_elements](#) (module), 167
[fobi.contrib.plugins.form_elements.fields.float.fobi_integration](#) (module), 168
 257
 fobi (module), 438
 fobi.admin (module), 373
 fobi.app (module), 377
 fobi.apps (module), 378
 fobi.base (module), 378
 fobi.conf (module), 397
 fobi.constants (module), 397
 fobi.context_processors (module), 397
 fobi.contrib (module), 354
 fobi.contrib.apps (module), 220
 fobi.contrib.apps.djangocms_integration (module), 165
 fobi.contrib.apps.djangocms_integration.apps (module), 164
 fobi.contrib.apps.djangocms_integration.conf (module), 164
 fobi.contrib.apps.djangocms_integration.defaults (module), 165
 fobi.contrib.apps.djangocms_integration.helper (module), 165
 fobi.contrib.apps.djangocms_integration.settings (module), 165
 fobi.contrib.apps.drf_integration (module), 216
 fobi.contrib.apps.drf_integration.apps (module), 210
 fobi.contrib.apps.drf_integration.base (module), 210
 fobi.contrib.apps.drf_integration.dynamic (module), 212
 fobi.contrib.apps.drf_integration.fields (module), 212
 fobi.contrib.apps.drf_integration.form_elements (module), 207
 fobi.contrib.apps.drf_integration.form_elements.base (module), 172
 fobi.contrib.apps.drf_integration.form_elements.contrib (module), 172
 fobi.contrib.apps.drf_integration.form_elements.contrib.apps.drf_integration (module), 166
 fobi.contrib.apps.drf_integration.form_elements.contrib.apps.drf_integration.apps (module), 172
 fobi.contrib.apps.drf_integration.form_elements.contrib.apps.drf_integration.form_elements (module), 165

544
Index

fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feincms_integration
 (module), 205 (module), 218
 fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feinapps_integration.apps
 (module), 204 (module), 216
 fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feibasms_integration.conf
 (module), 205 (module), 216
 fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feibms_integration.defaults
 (module), 205 (module), 216
 fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feincms_integration.helpers
 (module), 207 (module), 216
 fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feinapps_integration.settings
 (module), 205 (module), 216
 fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feibasms_integration.widgets
 (module), 206 (module), 217
 fobi.contrib.apps.drf_integration.form_elements.fobi_apps_feibms_integration.form_elements
 (module), 206 (module), 220
 fobi.contrib.apps.drf_integration.form_handlers.contrib.apps.mezzanine_integration.apps
 (module), 210 (module), 218
 fobi.contrib.apps.drf_integration.form_handlers.contrib.apps.mezzanine_integration.conf
 (module), 208 (module), 218
 fobi.contrib.apps.drf_integration.form_handlers.contrib.apps.mezzanine_integration.defaults
 (module), 207 (module), 219
 fobi.contrib.apps.drf_integration.form_handlers.contrib.apps.mezzanine_integration.helpers
 (module), 207 (module), 219
 fobi.contrib.apps.drf_integration.form_handlers.contrib.apps.mezzanine_integration.form_handlers
 (module), 207 (module), 219
 fobi.contrib.apps.drf_integration.form_handlers.contrib.reposts (module), 335
 (module), 209 fobi.contrib.plugins.form_elements (mod-
 fobi.contrib.apps.drf_integration.form_handlers.contrib.reposts.apps
 (module), 208 fobi.contrib.plugins.form_elements.content
 fobi.contrib.apps.drf_integration.form_handlers.contrib.reposts.base
 (module), 208 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.form_handlers.contrib.reposts.fobi_integration_form_handlers
 (module), 208 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.form_handlers.contrib.reposts.fobi_integration_form_handlers
 (module), 210 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.form_handlers.contrib.reposts.fobi_integration_form_handlers
 (module), 209 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.form_handlers.contrib.reposts.fobi_integration_form_handlers
 (module), 209 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.helpers (module), 221
 (module), 214 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.metadata (module), 222
 (module), 214 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.serializers (module), 223
 (module), 214 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.urls (module), 223
 (module), 215 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.utils (module), 226
 (module), 215 fobi.contrib.plugins.form_elements.content.content
 fobi.contrib.apps.drf_integration.views (module), 223
 (module), 215 fobi.contrib.plugins.form_elements.content.content

(module), 245

fobi.contrib.plugins.form_elements.fields.datetime (module), 243

fobi.contrib.plugins.form_elements.fields.datetimebase (module), 244

fobi.contrib.plugins.form_elements.fields.datetimefobiformelements (module), 244

fobi.contrib.plugins.form_elements.fields.datetimeforms (module), 245

fobi.contrib.plugins.form_elements.fields.datetimeintrib.plugins.form_elements.fields.file (module), 247

fobi.contrib.plugins.form_elements.fields.datetimeintribappplugins.form_elements.fields.file (module), 245

fobi.contrib.plugins.form_elements.fields.datetimeintribbaseplugins.form_elements.fields.float (module), 246

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.float (module), 246

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatbase (module), 247

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatfobiformelements (module), 247

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatforms (module), 249

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatintrib (module), 249

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatintribapp (module), 251

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatintribbase (module), 250

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatintribplugins (module), 250

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatintribpluginsbase (module), 251

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatintribpluginsfobiformelements (module), 253

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.floatintribpluginsforms (module), 251

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.integer (module), 252

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.integerbase (module), 252

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.integerfobiformelements (module), 253

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.integerforms (module), 253

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.integerintrib (module), 256

fobi.contrib.plugins.form_elements.fields.datetimeintribplugins.form_elements.fields.integerintribapp (module), 256

(module), 253

(module), 253

(module), 254

(module), 255

(module), 255

(module), 256

(module), 256

(module), 258

(module), 256

(module), 257

(module), 257

(module), 258

(module), 260

(module), 258

(module), 258

(module), 259

(module), 259

(module), 262

(module), 260

(module), 260

(module), 260

(module), 260

(module), 260

(module), 261

(module), 261

(module), 263

(module), 262

(module), 262

(module), 262

(module), 262

(module), 263
 fobi.contrib.plugins.form_elements.fields.fobi_address (module), 271
 (module), 265
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 272
 (module), 263
 (module), 272
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 273
 (module), 263
 (module), 273
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 273
 (module), 264
 (module), 273
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 275
 (module), 264
 (module), 275
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 273
 (module), 266
 (module), 273
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 274
 (module), 265
 (module), 274
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 274
 (module), 265
 (module), 274
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 275
 (module), 265
 (module), 275
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 278
 (module), 266
 (module), 278
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 275
 (module), 268
 (module), 275
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 275
 (module), 266
 (module), 275
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 276
 (module), 267
 (module), 276
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 276
 (module), 267
 (module), 276
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 276
 (module), 268
 (module), 276
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 277
 (module), 268
 (module), 277
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 278
 (module), 268
 (module), 278
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 280
 (module), 268
 (module), 280
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 278
 (module), 269
 (module), 278
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 278
 (module), 269
 (module), 278
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 279
 (module), 269
 (module), 279
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 279
 (module), 270
 (module), 279
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 279
 (module), 271
 (module), 279
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 280
 (module), 273
 (module), 280
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 280
 (module), 271
 (module), 280
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 282
 (module), 271
 (module), 282
 fobi.contrib.plugins.form_elements.fields.fobi_address_plugins (module), 282
 (module), 271
 (module), 282

| | |
|--------------|------------|
| Index | 549 |
|--------------|------------|

[illegible]

| | |
|--------------|------------|
| Index | 551 |
|--------------|------------|

- fobi.discover (module), 400
- fobi.dynamic (module), 401
- fobi.exceptions (module), 401
- fobi.form_importers (module), 403
- fobi.form_utils (module), 404
- fobi.forms (module), 404
- fobi.helpers (module), 409
- fobi.integration (module), 356
- fobi.integration.helpers (module), 354
- fobi.integration.processors (module), 355
- fobi.management (module), 357
- fobi.management.commands (module), 357
- fobi.management.commands.fobi_find_broken_entries (module), 356
- fobi.management.commands.fobi_sync_plugins (module), 356
- fobi.management.commands.fobi_update_plugins (module), 356
- fobi.migrations (module), 359
- fobi.migrations.0001_initial (module), 357
- fobi.migrations.0002_auto_20150912_1744 (module), 357
- fobi.migrations.0003_auto_20160517_1005 (module), 357
- fobi.migrations.0004_auto_20160906_1513 (module), 357
- fobi.migrations.0005_auto_20160908_1457 (module), 358
- fobi.migrations.0006_auto_20160911_1549 (module), 358
- fobi.migrations.0007_auto_20160926_1652 (module), 358
- fobi.migrations.0008_formwizardhandlerentry (module), 358
- fobi.migrations.0009_formwizardentry_widget_type (module), 358
- fobi.migrations.0010_formwizardhandler (module), 358
- fobi.migrations.0011_formentry_title (module), 359
- fobi.migrations.0012_auto_20161109_1550 (module), 359
- fobi.migrations.0013_formwizardentry_show_all_related_objects (module), 359
- fobi.migrations.0014_auto_20170516_1413 (module), 359
- fobi.models (module), 414
- fobi.reusable (module), 360
- fobi.reusable.markdown_widget (module), 360
- fobi.reusable.markdown_widget.helpers (module), 359
- fobi.reusable.markdown_widget.widgets (module), 360
- fobi.settings (module), 429
- fobi.templatetags (module), 362
- fobi.templatetags.fobi_tags (module), 360
- fobi.templatetags.future_compat (module), 362
- fobi.tests (module), 367
- fobi.tests.base (module), 362
- fobi.tests.constants (module), 363
- fobi.tests.core (module), 363
- fobi.tests.data (module), 363
- fobi.tests.helpers (module), 363
- fobi.tests.test_browser_build_dynamic_forms (module), 364
- fobi.tests.test_core (module), 365
- fobi.tests.test_drf_integration (module), 365
- fobi.tests.test_dynamic_forms (module), 366
- fobi.tests.test_feincms_integration (module), 366
- fobi.tests.test_form_importers_mailchimp (module), 366
- fobi.tests.test_sortable_dict (module), 367
- fobi.urls (module), 367
- fobi.urls.edit (module), 367
- fobi.urls.view (module), 367
- fobi.utils (module), 429
- fobi.validators (module), 432
- fobi.views (module), 432
- fobi.widgets (module), 438
- fobi.wizard (module), 373
- fobi.wizard.views (module), 373
- fobi.wizard.views.dynamic (module), 367
- fobi.wizard.views.views (module), 372
- FobiBrowserBuldDynamicFormsTest (class in fobi.tests.test_browser_build_dynamic_forms), 364
- FobiCoreTest (class in fobi.tests.test_core), 365
- FobiDataStructuresTest (class in fobi.tests.test_sortable_dict), 367
- FobiDjangoRestFrameworkIntegrationTests (class in fobi.tests.test_drf_integration), 365
- FobiDynamicFormsTest (class in fobi.tests.test_dynamic_forms), 366
- FobiFormEntryViewSet (class in fobi.contrib.apps.drf_integration.views), 215
- FobiFormWidget (class in fobi.contrib.apps.feincms_integration.widgets), 217
- FobiFormWidget.Meta (class in fobi.contrib.apps.feincms_integration.widgets), 217

| | |
|--|---|
| <code>attribute</code>), 353 | <code>tribute</code>), 425 |
| <code>form_edit_form_entry_option_class</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> <code>attribute</code>), 340 | <code>form_entry</code> (<code>fobi.models.ModelFormEntry</code> at- <code>tribute</code>), 427 |
| <code>form_edit_form_entry_option_class</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme</code> <code>attribute</code>), 344 | <code>form_entry_id</code> (<code>fobi.contrib.apps.feincms_integration.widgets.FobiFormWizardEntry</code> <code>attribute</code>), 217 |
| <code>form_edit_form_entry_option_class</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> <code>attribute</code>), 350 | <code>form_entry_id</code> (<code>fobi.models.AbstractPluginEntry</code> at- <code>tribute</code>), 419 |
| <code>form_edit_form_entry_option_class</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> <code>attribute</code>), 353 | <code>form_entry_id</code> (<code>fobi.models.FormFieldsetEntry</code> at- <code>tribute</code>), 424 |
| <code>form_edit_snippet_template_name</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme</code> <code>attribute</code>), 344 | <code>form_entry_id</code> (<code>fobi.models.ModelFormEntry</code> <code>attribute</code>), 428 |
| <code>form_edit_snippet_template_name</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> <code>attribute</code>), 353 | <code>form_entry_submitted()</code> (in module <code>fobi.views</code>), 436 |
| <code>form_element_checkbox_html_class</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> <code>attribute</code>), 340 | <code>form_entry_submitted_ajax_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> <code>attribute</code>), 340 |
| <code>form_element_checkbox_html_class</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme</code> <code>attribute</code>), 344 | <code>form_entry_submitted_ajax_template</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.</code> <code>attribute</code>), 344 |
| <code>form_element_checkbox_html_class</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> <code>attribute</code>), 350 | <code>form_entry_submitted_ajax_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> <code>attribute</code>), 350 |
| <code>form_element_checkbox_html_class</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> <code>attribute</code>), 353 | <code>form_entry_submitted_ajax_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> <code>attribute</code>), 353 |
| <code>form_element_html_class</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> <code>attribute</code>), 340 | <code>form_entry_submitted_template</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.</code> <code>attribute</code>), 344 |
| <code>form_element_html_class</code> (<code>fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme</code> <code>attribute</code>), 344 | <code>form_entry_submitted_template</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> <code>attribute</code>), 350 |
| <code>form_element_html_class</code> (<code>fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> <code>attribute</code>), 350 | <code>form_entry_submitted_template</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> <code>attribute</code>), 353 |
| <code>form_element_html_class</code> (<code>fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> <code>attribute</code>), 353 | <code>form_element_entry</code> (<code>fobi.models.ModelFormEntry</code> at- <code>tribute</code>), 421 |
| <code>form_entry</code> (<code>fobi.contrib.apps.feincms_integration.widgets.FobiFormWizardEntry</code> <code>attribute</code>), 217 | <code>form_element_entry_id</code> (<code>fobi.models.ModelFormEntry</code> at- <code>tribute</code>), 421 |
| <code>form_entry</code> (<code>fobi.contrib.plugins.form_handlers.db_store.models.SimpleFormDataEntry</code> <code>attribute</code>), 320 | <code>form_importer_ajax_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> <code>attribute</code>), 340 |
| <code>form_entry</code> (<code>fobi.models.AbstractPluginEntry</code> at- <code>tribute</code>), 419 | <code>form_importer_template</code> (<code>fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme</code> <code>attribute</code>), 340 |
| <code>form_entry</code> (<code>fobi.models.ModelFormEntry</code> at- <code>tribute</code>), 421 | <code>form_importers()</code> (in module <code>fobi.context_processors</code>), 397 |
| <code>form_entry</code> (<code>fobi.models.FormFieldsetEntry</code> at- <code>tribute</code>), 424 | |
| <code>form_entry</code> (<code>fobi.models.FormHandlerEntry</code> at- | |

form_list (fobi.contrib.plugins.form_importers.mailchimp_importer.widgets.SimpleTheme attribute), 334

form_list_container_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340

form_list_container_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 344

form_list_container_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350

form_list_container_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 353

form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340

form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350

form_non_field_and_hidden_errors_snippet_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 354

form_properties_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340

form_properties_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 344

form_properties_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350

form_properties_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 354

form_properties_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340

form_properties_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350

form_radio_element_html_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 344

form_radio_element_html_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 354

form_sent_get_param (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 217

form_sent_get_param (fobi.integration.processors.IntegrationProcessor attribute), 355

form_snippet_template_name (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340

form_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 344

form_snippet_template_name (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350

form_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 354

form_submit_button_text (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 217

form_template_name (fobi.DjangoCMSAdminStyleTheme attribute), 217

form_title (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 217

form_view_form_entry_option_class (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340

form_view_form_entry_option_class (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 344

form_view_form_entry_option_class (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350

form_view_form_entry_option_class (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 354

form_view_snippet_template_name (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme attribute), 344

form_view_snippet_template_name (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 354

form_wizard_ajax (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340

form_wizard_ajax (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350

form_wizard_entry (fobi.DjangoCMSAdminStyleTheme attribute), 320

form_wizard_entry (fobi.models.AbstractFormWizardPluginEntry attribute), 419

form_wizard_entry (fobi.models.FormWizardFormEntry attribute), 428

form_wizard_entry (fobi.models.FormWizardHandlerEntry attribute), 428

form_wizard_entry_id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormV attribute), 320

form_wizard_entry_id (fobi.DjangoCMSAdminStyleTheme attribute), 419

form_wizard_entry_id (fobi.models.FormWizardFormEntry attribute), 428

form_wizard_entry_submitted() (in module

[fobi.views](#)), 436
[form_wizard_properties_snippet_template_name](#) (in module [fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#)), 340
[form_wizard_properties_snippet_template_name](#) (in module [fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#)), 350
[form_wizard_properties_snippet_template_name](#) (in module [fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#)), 354
[form_wizard_snippet_template_name](#) (in module [fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#)), 340
[form_wizard_snippet_template_name](#) (in module [fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#)), 350
[form_wizard_snippet_template_name](#) (in module [fobi.contrib.themes.simple.fobi_themes.SimpleTheme](#)), 354
[form_wizard_template](#) (in module [fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#)), 340
[form_wizards_dashboard\(\)](#) (in module [fobi.views](#)), 436
[form_wizards_dashboard_template](#) (in module [fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme](#)), 340
[form_wizards_dashboard_template](#) (in module [fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme](#)), 350
[formatted_saved_data\(\)](#) (in module [fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormHandler](#)), 319
[FormCallback](#) (class in [fobi.base](#)), 385
[FormCallbackError](#), 402
[FormCallbackRegistry](#) (class in [fobi.base](#)), 385
[FormElement](#) (class in [fobi.models](#)), 415
[FormElement.DoesNotExist](#), 416
[FormElement.MultipleObjectsReturned](#), 416
[FormElementAdmin](#) (class in [fobi.admin](#)), 374
[FormElementEntry](#) (class in [fobi.models](#)), 420
[FormElementEntry.DoesNotExist](#), 420
[FormElementEntry.MultipleObjectsReturned](#), 420
[formelemententry_set](#) (in module [fobi.models.FormEntry](#)), 422
[formelemententry_set](#) (in module [fobi.models.FormFieldsetEntry](#)), 424
[FormElementEntryAdmin](#) (class in [fobi.admin](#)), 374
[FormElementEntryAdmin.Meta](#) (class in [fobi.admin](#)), 374
[FormElementEntryForm](#) (class in [fobi.forms](#)), 405
[FormElementEntryForm.Meta](#) (class in [fobi.forms](#)), 405
[FormElementEntryFormSet](#) (in module [fobi.forms](#)), 405
[FormElementEntryInlineAdmin](#) (class in [fobi.admin](#)), 374
[FormElementPlugin](#) (class in [fobi.base](#)), 385
[FormElementPluginDataStorage](#) (class in [fobi.base](#)), 386
[FormElementPluginDoesNotExist](#), 402
[FormElementPluginError](#), 402
[FormElementPluginRegistry](#) (class in [fobi.base](#)), 387
[FormElementPluginWidget](#) (class in [fobi.base](#)), 387
[FormElementPluginWidgetRegistry](#) (class in [fobi.base](#)), 387
[FormEntry](#) (class in [fobi.models](#)), 421
[FormEntry.DoesNotExist](#), 421
[FormEntry.MultipleObjectsReturned](#), 421
[FormEntryAdmin](#) (class in [fobi.admin](#)), 375
[FormEntryAdmin.Meta](#) (class in [fobi.admin](#)), 375
[FormEntryForm](#) (class in [fobi.forms](#)), 405
[FormEntryForm.Meta](#) (class in [fobi.forms](#)), 405
[FormEntrySerializer](#) (class in [fobi.contrib.apps.drf_integration.serializers](#)), 424
[FormEntrySerializer.Meta](#) (class in [fobi.contrib.apps.drf_integration.serializers](#)), 424
[FormFieldPlugin](#) (class in [fobi.base](#)), 387
[FormFieldsetEntry](#) (class in [fobi.models](#)), 424
[FormFieldsetEntry.DoesNotExist](#), 424
[FormFieldsetEntry.MultipleObjectsReturned](#), 424
[formfieldsetentry_set](#) (in module [fobi.models.FormEntry](#)), 422
[FormFieldsetEntryAdmin](#) (class in [fobi.admin](#)), 375
[FormFieldsetEntryAdmin.Meta](#) (class in [fobi.admin](#)), 375
[FormFieldsetEntryForm](#) (class in [fobi.forms](#)), 406
[FormFieldsetEntryForm.Meta](#) (class in [fobi.forms](#)), 406
[FormHandler](#) (class in [fobi.models](#)), 416
[FormHandler.DoesNotExist](#), 417
[FormHandler.MultipleObjectsReturned](#), 417
[FormHandlerAdmin](#) (class in [fobi.admin](#)), 375
[FormHandlerEntry](#) (class in [fobi.models](#)), 425
[FormHandlerEntry.DoesNotExist](#), 425
[FormHandlerEntry.MultipleObjectsReturned](#), 425
[formhandlerentry_set](#) (in module [fobi.models.FormEntry](#)), 422
[FormHandlerEntryAdmin](#) (class in [fobi.admin](#)), 376

FormHandlerEntryAdmin.Meta (class in formwizardformentry_set
 fobi.admin), 376 (fobi.models.FormWizardEntry attribute),
 FormHandlerEntryForm (class in *fobi.forms*), 406 425
 FormHandlerEntryForm.Meta (class in FormWizardFormEntryForm (class in *fobi.forms*),
 fobi.forms), 406 407
 FormHandlerEntryInlineAdmin (class in FormWizardFormEntryForm.Meta (class in
 fobi.admin), 376 *fobi.forms*), 407
 FormHandlerForm (class in *fobi.forms*), 406 FormWizardFormEntryFormSet (in module
 fobi.forms), 408
 FormHandlerForm.Meta (class in *fobi.forms*), 407
 FormHandlerPlugin (class in *fobi.base*), 387 FormWizardFormEntryInlineAdmin (class in
 fobi.admin), 377
 FormHandlerPluginDataStorage (class in *fobi.base*), 388 FormWizardHandler (class in *fobi.models*), 417
 FormHandlerPluginDoesNotExist, 402 FormWizardHandler.DoesNotExist, 418
 FormHandlerPluginError, 402 FormWizardHandler.MultipleObjectsReturned,
 418
 FormHandlerPluginRegistry (class in *fobi.base*), 388 FormWizardHandlerAdmin (class in *fobi.admin*),
 377
 FormHandlerPluginWidget (class in *fobi.base*), 388 FormWizardHandlerEntry (class in *fobi.models*),
 428
 FormHandlerPluginWidgetRegistry (class in *fobi.base*), 388 FormWizardHandlerEntry.DoesNotExist, 428
 FormImporterPluginRegistry (class in FormWizardHandlerEntry.MultipleObjectsReturned,
 fobi.form_importers), 403 428
 FormImportersMailchimpTest (class in formwizardhandlerentry_set
 fobi.tests.test_form_importers_mailchimp), (fobi.models.FormWizardEntry attribute),
 366 426
 FormPluginError, 402 FormWizardHandlerEntryForm (class in
 fobi.forms), 408
 forms_list_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3ThemeHandlerEntryForm.Meta (class in
 attribute), 340 *fobi.forms*), 408
 forms_list_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.djangocms_admin_style_theme
 attribute), 344 FormWizardHandlerEntryInlineAdmin (class
 in *fobi.admin*), 376
 forms_list_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5ThemeHandlerEntryForm (class in
 attribute), 350 *fobi.forms*), 408
 forms_list_template (fobi.contrib.themes.simple.fobi_themes.SimpleThemeHandlerEntryForm (class in
 attribute), 354 *fobi.forms*), 408
 FormWizardEntry (class in *fobi.models*), 425 FormWizardHandlerEntryInlineAdmin (class
 in *fobi.admin*), 376 FormWizardHandlerPlugin (class in *fobi.base*),
 388
 FormWizardEntry.DoesNotExist, 425 FormWizardHandlerPluginDataStorage (class
 in *fobi.base*), 389
 FormWizardEntry.MultipleObjectsReturned, 425 FormWizardHandlerPluginDoesNotExist, 402
 FormWizardHandlerPluginRegistry (class in
 fobi.base), 389
 FormWizardEntryAdmin (class in *fobi.admin*), 376 FormWizardHandlerPluginWidget (class in
 fobi.admin), 376 *fobi.base*), 389
 FormWizardEntryAdmin.Meta (class in *fobi.admin*), 376 FormWizardHandlerPluginWidgetRegistry
 (class in *fobi.base*), 390
 FormWizardEntryForm (class in *fobi.forms*), 407 FormWizardView (class in *fobi.views*), 437
 FormWizardEntryForm.Meta (class in *fobi.forms*), 407 Foundation5Theme (class in
 fobi.contrib.themes.foundation5.fobi_themes),
 349
 FormWizardFormEntry (class in *fobi.models*), 427
 FormWizardFormEntry.DoesNotExist, 427
 FormWizardFormEntry.MultipleObjectsReturned, 427
 formwizardformentry_set (fobi.models.FormEntry attribute), 422

G

generate_ticks() (in module
 fobi.contrib.plugins.form_elements.fields.slider.helpers),
 296
 get() (*fobi.base.BaseRegistry* method), 383
 get() (*fobi.wizard.views.dynamic.DynamicNamedUrlWizardView*
 method), 371

[get\(\)](#) (*fobi.wizard.views.dynamic.DynamicWizardView* method), 368
[get\(\)](#) (*fobi.wizard.views.views.CookieWizardView* method), 373
[get\(\)](#) (*fobi.wizard.views.views.SessionWizardView* method), 373
[get\(\)](#) (*fobi.wizard.views.views.WizardView* method), 372
[get_absolute_url\(\)](#) (*fobi.models.FormEntry* method), 423
[get_absolute_url\(\)](#) (*fobi.models.FormWizardEntry* method), 426
[get_all_cleaned_data\(\)](#) (*fobi.wizard.views.dynamic.DynamicWizardView* method), 368
[get_allowed_form_element_plugin_uids\(\)](#) (in module *fobi.utils*), 429
[get_allowed_form_handler_plugin_uids\(\)](#) (in module *fobi.utils*), 429
[get_allowed_form_wizard_handler_plugin_uids\(\)](#) (in module *fobi.utils*), 430
[get_allowed_plugin_uids\(\)](#) (in module *fobi.utils*), 430
[get_app_label_and_model_name\(\)](#) (in module *fobi.helpers*), 410
[get_base_url\(\)](#) (*fobi.contrib.plugins.form_handlers.mail_mixin.MailHandlerMixin* method), 331
[get_callbacks\(\)](#) (*fobi.base.FormCallbackRegistry* method), 385
[get_callbacks\(\)](#) (*fobi.base.IntegrationFormCallbackRegistry* method), 394
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_base.CheckboxSelectMultipleInputPlugin* method), 239
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_elements.CheckboxSelectMultipleInputPlugin* method), 240
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.range_select_base.RangeSelectInputPlugin* method), 271
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements.RangeSelectInputPlugin* method), 272
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.select_base.SelectInputPlugin* method), 276
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.SelectInputPlugin* method), 277
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.select_multiple_base.SelectMultipleInputPlugin* method), 283
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_elements.SelectMultipleInputPlugin* method), 284
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements.SelectMultipleWithMaxInputPlugin* method), 290
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements.SelectMultipleWithMaxInputPlugin* method), 292
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.slider_base.SliderInputPlugin* method), 294
[get_choices\(\)](#) (*fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements.SliderInputPlugin* method), 295
[get_cleaned_data\(\)](#) (in module *fobi.contrib.apps.drf_integration.base*), 210
[get_cleaned_data_for_step\(\)](#) (*fobi.wizard.views.dynamic.DynamicWizardView* method), 368
[get_cloned_plugin_data\(\)](#) (*fobi.base.BasePlugin* method), 379
[get_context_data\(\)](#) (*fobi.integration.processors.IntegrationProcessor* method), 355
[get_context_data\(\)](#) (*fobi.views.FormWizardView* method), 437
[get_context_data\(\)](#) (*fobi.wizard.views.dynamic.DynamicNamedUrlWizardView* method), 371
[get_context_data\(\)](#) (*fobi.wizard.views.dynamic.DynamicWizardView* method), 368
[get_prop_filter\(\)](#) (in module *fobi.contrib.plugins.form_elements.content.content_image.helpers*), 223
[get_custom_actions\(\)](#) (*fobi.base.FormHandlerPlugin* method), 387
[get_custom_actions\(\)](#) (*fobi.base.FormWizardHandlerPlugin* method), 389
[get_custom_field_instances\(\)](#) (*fobi.base.FormElementPlugin* method), 385
[get_custom_field_instances\(\)](#) (*fobi.contrib.apps.drf_integration.form_elements.content.content_image.helpers* method), 166
[get_custom_field_instances\(\)](#) (*fobi.contrib.apps.drf_integration.form_elements.content.content_image.helpers* method), 167

| | |
|---|---|
| (fobi.contrib.apps.drf_integration.form_elements.content.col | (fobi.contrib.apps.drf_integration.form_elements.fields RichTextP |
| method), 169 | method), 179 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.content.col | (fobi.contrib.apps.drf_integration.form_elements.fields.email.base |
| method), 170 | method), 180 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.content.col | (fobi.contrib.apps.drf_integration.form_elements.fields.Pluginfobi |
| method), 170 | method), 181 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.content.col | (fobi.contrib.apps.drf_integration.form_elements.fields.file.base.F |
| method), 171 | method), 181 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.content.col | (fobi.contrib.apps.drf_integration.form_elements.fields.FilePlugin |
| method), 171 | method), 182 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.boole | (fobibase.BaseAppSelfIntegration.form_elements.fields.float.base. |
| method), 172 | method), 183 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.boole | (fobibase.BaseAppSelfIntegration.form_elements.fields.float.base. |
| method), 172 | method), 183 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.check | (fobi.contrib.apps.drf_integration.form_elements.fields.CheckedMultipleSubPlugina |
| method), 173 | method), 184 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.check | (fobi.contrib.apps.drf_integration.form_elements.fields.CheckedOfS |
| method), 174 | method), 184 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.date | (fobibase.BaseAppSelfIntegration.form_elements.fields.input.base |
| method), 175 | method), 185 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.date | (fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi |
| method), 175 | method), 185 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.date | (fobibase.BaseAppSelfIntegration.form_elements.fields.integer.ba |
| method), 176 | method), 186 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.date | (fobibase.BaseAppSelfIntegration.form_elements.fields.IntegerSelec |
| method), 176 | method), 186 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.date | (fobibase.BaseAppSelfIntegration.form_elements.fields.ip_addres |
| method), 177 | method), 187 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.date | (fobibase.BaseAppSelfIntegration.form_elements.fields.null_boole |
| method), 177 | method), 187 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.decim | (fobibase.BaseAppSelfIntegration.form_elements.fields.null_boole |
| method), 178 | method), 188 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.decim | (fobibase.BaseAppSelfIntegration.form_elements.fields.null_boole |
| method), 178 | method), 188 |
| get_custom_field_instances() | get_custom_field_instances() |
| (fobi.contrib.apps.drf_integration.form_elements.fields.durat | (fobibase.BaseAppSelfIntegration.form_elements.fields.password. |
| method), 179 | method), 189 |
| get_custom_field_instances() | get_custom_field_instances() |

(in module `fobi.helpers`), 410

`get_form_element_plugin_widget()` (in module `fobi.base`), 390

`get_form_entry_for_step()` (`fobi.views.FormWizardView` method), 437

`get_form_field_instances()` (`fobi.base.FormElementPlugin` method), 386

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.base.ContentImagePlugin` method), 220

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 222

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 223

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 224

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 226

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 227

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 229

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 230

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 232

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 233

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 234

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.content.content_image.fobi_form_element_plugins.content_image.fobi_form_element_plugins` method), 235

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.boolean.base.BooleanPlugin` method), 237

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_element_plugins.boolean.fobi_form_element_plugins` method), 238

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_element_plugins.checkbox_select_multiple.fobi_form_element_plugins` method), 239

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi_form_element_plugins.checkbox_select_multiple.fobi_form_element_plugins` method), 240

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin` method), 242

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.date.fobi_form_element_plugins.date.fobi_form_element_plugins` method), 242

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.date_drop_down.base.DateDropDownPlugin` method), 244

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.date_drop_down.fobi_form_element_plugins.date_drop_down.fobi_form_element_plugins` method), 244

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.datetime.base.DateTimePlugin` method), 246

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.datetime.fobi_form_element_plugins.datetime.fobi_form_element_plugins` method), 246

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.decimal.base.DecimalPlugin` method), 248

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.decimal.fobi_form_element_plugins.decimal.fobi_form_element_plugins` method), 248

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.duration.base.DurationPlugin` method), 250

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.duration.fobi_form_element_plugins.duration.fobi_form_element_plugins` method), 250

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.email.base.EmailInputPlugin` method), 252

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.email.fobi_form_element_plugins.email.fobi_form_element_plugins` method), 252

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin` method), 254

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.file.fobi_form_element_plugins.file.fobi_form_element_plugins` method), 255

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.float.base.FloatInputPlugin` method), 257

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.float.fobi_form_element_plugins.float.fobi_form_element_plugins` method), 257

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.hidden.base.HiddenInputPlugin` method), 259

`get_form_field_instances()` (`fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_element_plugins.hidden.fobi_form_element_plugins` method), 259

| | |
|---|---|
| <code>(fobi.contrib.plugins.form_elements.fields.hidden.fobi_form_element.plugin_info.PluginInfo) method), 259</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_model_object.base.SelectModelObjectPlugin) method), 277</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.input.base.InputPlugin) method), 260</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_model_object.base.SelectModelObjectPlugin) method), 278</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.input.fobi_form_element.plugin_info.PluginInfo) method), 261</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_model_object.base.SelectModelObjectPlugin) method), 279</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.integer.base.IntegerInputPlugin) method), 262</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base.SelectMpttModelObjectPlugin) method), 280</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.integer.fobi_form_element.plugin_info.PluginInfo) method), 262</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.base.SelectMpttModelObjectPlugin) method), 281</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.ip_address.base.IPAddressInputPlugin) method), 264</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple.base.SelectMultiplePlugin) method), 283</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.ip_address.fobi_form_element.plugin_info.PluginInfo) method), 264</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_form_element.plugin_info.PluginInfo) method), 284</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.null_boolean.base.NullBooleanSelectPlugin) method), 265</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.base.SelectMultipleModelObjectPlugin) method), 286</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.null_boolean.fobi_form_element.plugin_info.PluginInfo) method), 266</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.base.SelectMultipleModelObjectPlugin) method), 287</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.password.base.PasswordInputPlugin) method), 267</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_object.base.SelectMultipleMpttModelObjectPlugin) method), 288</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.password.fobi_form_element.plugin_info.PluginInfo) method), 267</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_object.base.SelectMultipleMpttModelObjectPlugin) method), 289</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.radio.base.RadioInputPlugin) method), 269</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple_with_nested_model_object.base.SelectMultipleWithNestedModelObjectPlugin) method), 290</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.radio.fobi_form_element.plugin_info.PluginInfo) method), 269</code> | <code>(fobi.contrib.plugins.form_elements.fields.select_multiple_with_nested_model_object.base.SelectMultipleWithNestedModelObjectPlugin) method), 292</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.range_select.base.RangeSelectPlugin) method), 271</code> | <code>(fobi.contrib.plugins.form_elements.fields.slider.base.SliderInputPlugin) method), 294</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_element.plugin_info.PluginInfo) method), 272</code> | <code>(fobi.contrib.plugins.form_elements.fields.slider.fobi_form_element.plugin_info.PluginInfo) method), 295</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.regex.base.RegexInputPlugin) method), 274</code> | <code>(fobi.contrib.plugins.form_elements.fields.slug.base.SlugInputPlugin) method), 297</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_element.plugin_info.PluginInfo) method), 274</code> | <code>(fobi.contrib.plugins.form_elements.fields.slug.fobi_form_element.plugin_info.PluginInfo) method), 297</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select.base.SelectInputPlugin) method), 276</code> | <code>(fobi.contrib.plugins.form_elements.fields.text.base.TextInputPlugin) method), 299</code> |
| <code>get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.select_model_object.fobi_form_element.plugin_info.PluginInfo) method), 276</code> | <code>(fobi.contrib.plugins.form_elements.fields.text.fobi_form_element.plugin_info.PluginInfo) method), 299</code> |

(fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin method), 299

(fobi.wizard.views.dynamic.DynamicWizardView method), 369

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.textarea.base.TextareaPlugin method), 300

(fobi.wizard.views.dynamic.DynamicWizardView method), 369

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements.TextareaPlugin method), 301

(fobi.contrib.plugins.form_importers.mailchimp_importer.views.M method), 334

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin method), 302

(fobi.wizard.views.dynamic.DynamicWizardView method), 369

get_form_field_instances() (fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin method), 302

(fobi.wizard.views.dynamic.DynamicWizardView method), 369

get_form_field_instances() get_form_prefix() (fobi.wizard.views.dynamic.DynamicWizardView method), 369

(fobi.contrib.plugins.form_elements.fields.url.base.URLInputPlugin method), 304

get_form_field_instances() get_form_step_data() (fobi.wizard.views.dynamic.DynamicWizardView method), 369

(fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin method), 304

get_form_field_instances() get_form_step_files() (fobi.wizard.views.dynamic.DynamicWizardView method), 369

(fobi.contrib.plugins.form_elements.security.captcha.base.CaptchaInputPlugin method), 306

get_form_field_instances() get_form_template_choices() (in module fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaInputPlugin integration.helpers), 165

(fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaInputPlugin method), 306

get_form_field_instances() get_form_template_choices() (in module fobi.contrib.plugins.form_elements.security.honeypot.base.HoneypotInputPlugin integration.helpers), 216

(fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin method), 308

get_form_field_instances() get_form_template_choices() (in module fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin integration.helpers), 219

(fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin method), 309

get_form_field_instances() get_form_template_name() (fobi.contrib.plugins.form_elements.security.recaptcha.base.ReCaptchaInputPlugin method), 310

(fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements.ReCaptchaInputPlugin method), 310

get_form_field_instances() get_form_template_name_display() (fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements.ReCaptchaInputPlugin method), 310

(fobi.contrib.plugins.form_elements.test.dummy.base.DummyPlugin method), 312

get_form_field_instances() get_form_wizard_handler_plugin_widget() (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin method), 312

(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin method), 312

get_full_name() (fobi.helpers.StrippedUser method), 413

get_form_field_instances() (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin method), 312

get_form_field_instances() (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin method), 312

get_form_field_type() (in module fobi.templatetags.fobi_tags), 361

get_form_handler_plugin_widget() (in module fobi.base), 390

get_form_hidden_fields_errors() (in module fobi.templatetags.fobi_tags), 361

get_form_importer_plugin_uids() (in module fobi.form_importers), 404

get_form_importer_plugin_urls() (in module fobi.form_importers), 404

get_ignorable_field_names() (fobi.views.FormWizardView method), 437

get_ignorable_form_values() (in module fobi.helpers), 410

get_initial() (fobi.contrib.plugins.form_elements.fields.range_select method), 271

get_initial() (fobi.contrib.plugins.form_elements.fields.range_select method), 272

get_initial() (fobi.contrib.plugins.form_elements.fields.slider.base.S method), 294

`get_initial()` (*fobi.contrib.plugins.form_elements.fields.slider.fobi.contrib.plugins.form_elements.SliderInputPlugin method*), 295
`get_initial_wizard_data()` (*fobi.views.FormWizardView method*), 437
`get_initial_wizard_data()` (*fobi.wizard.views.dynamic.DynamicWizardView method*), 369
`get_initialised_create_form()` (*fobi.base.BasePlugin method*), 380
`get_initialised_create_form_or_404()` (*fobi.base.BasePlugin method*), 380
`get_initialised_edit_form()` (*fobi.base.BasePlugin method*), 380
`get_initialised_edit_form_or_404()` (*fobi.base.BasePlugin method*), 380
`get_initkwargs()` (*fobi.wizard.views.dynamic.DynamicNamedFormWizardView class method*), 371
`get_initkwargs()` (*fobi.wizard.views.dynamic.DynamicWizardView class method*), 369
`get_instance()` (*fobi.base.BasePlugin method*), 380
`get_login_required_template_name()` (*fobi.integration.processors.IntegrationProcessor method*), 355
`get_model_name_for_object()` (in module *fobi.helpers*), 410
`get_next_by_created()` (*fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormEntry method*), 319
`get_next_by_created()` (*fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry method*), 320
`get_next_by_created()` (*fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardEntry method*), 320
`get_next_step()` (*fobi.wizard.views.dynamic.DynamicWizardView method*), 370
`get_object()` (*fobi.contrib.apps.drf_integration.views.FobiFormEntryView method*), 215
`get_or_create_admin_user()` (in module *fobi.tests.helpers*), 364
`get_ordered_form_handlers()` (in module *fobi.base*), 390
`get_ordered_form_wizard_handlers()` (in module *fobi.base*), 391
`get_origin_kwargs_update_func_results()` (*fobi.base.FormElementPlugin method*), 386
`get_origin_return_func_results()` (*fobi.base.FormElementPlugin method*), 386
`get_plugin()` (*fobi.models.BaseAbstractPluginEntry method*), 420
`get_plugin_data()` (*fobi.base.BasePluginForm method*), 383
`get_plugin_form_data()` (*fobi.base.BasePlugin method*), 380
`get_plugin_widget()` (in module *fobi.base*), 391
`get_prefix()` (*fobi.wizard.views.dynamic.DynamicWizardView method*), 370
`get_prev_step()` (*fobi.wizard.views.dynamic.DynamicWizardView method*), 370
`get_previous_by_created()` (*fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormEntry method*), 319
`get_previous_by_created()` (*fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry method*), 320
`get_previous_by_created()` (*fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardEntry method*), 321
`get_redirect_url()` (*fobi.integration.processors.IntegrationProcessor method*), 355
`get_processed_form_data()` (in module *fobi.base*), 391
`get_processed_form_wizard_data()` (in module *fobi.base*), 391
`get_processed_serializer_data()` (in module *fobi.contrib.apps.drf_integration.base*), 211
`get_queryset()` (*fobi.admin.BasePluginModelAdmin method*), 373
`get_queryset()` (*fobi.admin.SavedFormEntryAdmin method*), 374
`get_queryset()` (*fobi.admin.FormHandlerEntryAdmin method*), 374
`get_queryset()` (*fobi.contrib.apps.drf_integration.views.FobiFormEntryView method*), 215
`get_queryset()` (*fobi.contrib.plugins.form_elements.fields.select_mode method*), 278
`get_queryset()` (*fobi.contrib.plugins.form_elements.fields.select_mode method*), 279
`get_queryset()` (*fobi.contrib.plugins.form_elements.fields.select_mode method*), 286
`get_queryset()` (*fobi.contrib.plugins.form_elements.fields.select_mode method*), 287
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_in method*), 221
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_in method*), 222
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_in method*), 223
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_in method*), 224
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_in method*), 226
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_in method*), 227
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_in method*), 229

`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin*
method), 230 `get_registered_plugins()` (*fobi.models.BaseAbstractPluginEntry* *method*), 420
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin*
method), 232 `get_registered_plugins()` (*fobi.models.BaseAbstractPluginEntry* *method*), 425
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin*
method), 233 `get_registered_plugins()` (*fobi.models.BaseAbstractPluginEntry* *method*), 428
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin*
method), 234 `get_registered_plugins()` (*fobi.models.BaseAbstractPluginEntry* *method*), 428
`get_raw_data()` (*fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin*
method), 235 `get_registered_plugins()` (*fobi.models.BaseAbstractPluginEntry* *method*), 428
`get_registered_form_callbacks()` (*in module fobi.base*), 391 `get_registered_plugins()` (*in module fobi.base*), 393
`get_registered_form_element_plugin_uids()` (*in module fobi.base*), 391 `get_registered_theme_uids()` (*in module fobi.base*), 393
`get_registered_form_element_plugins()` (*in module fobi.base*), 391 `get_registered_themes()` (*in module fobi.base*), 393
`get_registered_form_element_plugins_grouped()` (*in module fobi.base*), 391 `get_registry()` (*fobi.models.BaseAbstractPluginEntry*
method), 420
`get_registered_form_handler_plugin_uids()` (*in module fobi.base*), 392 `get_registry()` (*fobi.models.BaseAbstractPluginEntry*
method), 421
`get_registered_form_handler_plugins()` (*in module fobi.base*), 392 `get_registry()` (*fobi.models.BaseAbstractPluginEntry*
method), 425
`get_registered_form_wizard_handler_plugin_uids()` (*in module fobi.base*), 392 `get_registry()` (*fobi.models.BaseAbstractPluginEntry*
method), 429
`get_registered_form_wizard_handler_plugins()` (*in module fobi.base*), 392 `get_rendered_data()`
(*fobi.contrib.plugins.form_handlers.mail.mixins.MailHandlerMixin* *method*), 331
`get_registered_integration_form_element_plugin_uids()` (*in module fobi.base*), 392 `get_rendered_image()`
(*fobi.contrib.plugins.form_elements.content.content_image.base.ContentImagePlugin* *method*), 221
`get_registered_integration_form_element_plugins_grouped()` (*in module fobi.base*), 392 `get_rendered_image()`
(*fobi.contrib.plugins.form_elements.content.content_image.fobi_f* *method*), 222
`get_registered_integration_form_handler_plugin_uids()` (*in module fobi.base*), 392 `get_rendered_image()`
(*fobi.contrib.plugins.form_elements.content.content_image_url.base.ContentImageURLPlugin* *method*), 224
`get_registered_integration_form_handler_plugins_grouped()` (*in module fobi.base*), 393 `get_rendered_image()`
(*fobi.contrib.plugins.form_elements.content.content_image_url.fobi_f* *method*), 224
`get_registered_models()` (*in module fobi.helpers*), 410 `get_rendered_text()`
(*fobi.contrib.plugins.form_elements.content.content_markdown.base.ContentMarkdownPlugin* *method*), 226
`get_registered_plugin_uids()` (*in module fobi.base*), 393 `get_rendered_text()`
(*fobi.contrib.plugins.form_elements.content.content_markdown.fobi_f* *method*), 227
`get_registered_plugins()` (*fobi.models.AbstractPluginModel* *method*), 415 `get_rendered_text()`
(*fobi.contrib.plugins.form_elements.content.content_richtext.base.ContentRichTextPlugin* *method*), 230
`get_registered_plugins()` (*fobi.models.BaseAbstractPluginEntry* *method*), 420 `get_rendered_text()`
(*fobi.contrib.plugins.form_elements.content.content_richtext.fobi_f* *method*), 230
`get_registered_plugins()` (*fobi.models.FormElement* *method*), 416 `get_rendered_text()`
(*fobi.contrib.plugins.form_elements.content.content_richtext.fobi_f* *method*), 230
`get_registered_plugins()` (*fobi.models.FormElementEntry* *method*), 421 `get_rendered_text()`
(*fobi.contrib.plugins.form_elements.content.content_text.base.ContentTextPlugin* *method*), 232
`get_registered_plugins()` (*fobi.contrib.plugins.form_elements.content.content_text.fobi_f* *method*), 232

[get_rendered_text\(\)](#) (fobi.contrib.plugins.form_elements.content.content_text.fobi_form_elements.ContentTextPlugin module method), 233 [276](#)
[get_rendered_video\(\)](#) (fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.ContentVideoPlugin module method), 234 [279](#)
[get_rendered_video\(\)](#) (fobi.contrib.plugins.form_elements.content.content_video.fobi_form_elements.ContentVideoPlugin module method), 235 [281](#)
[get_select_field_choices\(\)](#) (in module fobi.helpers), 410 [get_setting\(\)](#) (in module fobi.contrib.apps.drf_integration.views.FobiFormEntryView), 286
[get_serializer\(\)](#) (fobi.contrib.apps.drf_integration.views.FobiFormEntryView module method), 215 [286](#)
[get_serializer_class\(\)](#) (fobi.contrib.apps.drf_integration.views.FobiFormEntryView module method), 215 [289](#)
[get_serializer_class\(\)](#) (in module fobi.contrib.apps.drf_integration.utils), 215 [get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.conf), 291
[get_setting\(\)](#) (in module fobi.conf), 397 [291](#)
[get_setting\(\)](#) (in module fobi.contrib.apps.djangocms_integration.conf), 164 [294](#)
[get_setting\(\)](#) (in module fobi.contrib.apps.feincms_integration.conf), 216 [308](#)
[get_setting\(\)](#) (in module fobi.contrib.apps.mezzanine_integration.conf), 218 [317](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.content.content_image.conf), 221 [328](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.content.content_image.conf), 224 [413](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.content.content_text.conf), 232 [370](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.content.content_video.conf), 235 [372](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.fobi.contrib.apps.feincms_integration.conf), 239 [372](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.fields.file.conf), 254 [372](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.fields.radio.conf), 269 [372](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.fields.range_select.conf), 271 [372](#)
[get_setting\(\)](#) (in module fobi.contrib.plugins.form_elements.fields.select.conf), 354 [372](#)

`get_template_names()`
 (*fobi.form_importers.BaseFormImporter*
 method), 403
`get_theme()` (*in module fobi.base*), 394
`get_updated_plugin_data()`
 (*fobi.base.BasePlugin method*), 381
`get_url()` (*fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer*), 166
 method), 214
`get_urls()` (*fobi.admin.FormElementAdmin method*),
 374
`get_urls()` (*fobi.admin.FormHandlerAdmin method*),
 375
`get_urls()` (*fobi.admin.FormWizardHandlerAdmin*
 method), 377
`get_user_form_element_plugin_uids()` (*in*
 module fobi.utils), 430
`get_user_form_element_plugins()` (*in mod-*
 ule fobi.utils), 430
`get_user_form_element_plugins_grouped()`
 (*in module fobi.utils*), 430
`get_user_form_field_plugin_uids()` (*in*
 module fobi.utils), 430
`get_user_form_handler_plugin_uids()` (*in*
 module fobi.utils), 430
`get_user_form_handler_plugins()` (*in mod-*
 ule fobi.utils), 430
`get_user_form_handler_plugins_grouped()`
 (*in module fobi.utils*), 430
`get_user_form_wizard_handler_plugin_uids()`
 (*in module fobi.utils*), 430
`get_user_form_wizard_handler_plugins()`
 (*in module fobi.utils*), 430
`get_user_form_wizard_handler_plugins_grouped()`
 (*in module fobi.utils*), 430
`get_user_plugin_uids()` (*in module fobi.utils*),
 431
`get_user_plugins()` (*in module fobi.utils*), 431
`get_user_plugins_grouped()` (*in module*
 fobi.utils), 431
`get_username()` (*fobi.helpers.StrippedUser*
 method), 413
`get_widget()` (*fobi.base.BasePlugin method*), 381
`get_wizard()` (*fobi.form_importers.BaseFormImporter*
 method), 403
`get_wizard_files_upload_dir()` (*in module*
 fobi.utils), 431
`get_wizard_form_field_value_from_post()`
 (*in module fobi.helpers*), 410
`get_wizard_form_field_value_from_request()`
 (*in module fobi.helpers*), 411
`get_wizard_form_field_value_from_session()`
 (*in module fobi.helpers*), 411
`get_wizard_type_display()`
 (*fobi.models.FormWizardEntry method*),
 426
`graceful_export()`
 (*fobi.contrib.plugins.form_handlers.db_store.helpers.DataExport*
 method), 318
`group` (*fobi.base.BasePlugin attribute*), 381
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_in-*
 tegration_form_element attribute), 166
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_in-*
 tegration_form_element attribute), 166
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_in-*
 tegration_form_element attribute), 167
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_in-*
 tegration_form_element attribute), 167
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_m-*
 odel_form_element attribute), 168
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_m-*
 odel_form_element attribute), 168
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_r-*
 esource_form_element attribute), 169
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_r-*
 esource_form_element attribute), 169
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_te-*
 mplate_form_element attribute), 170
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_te-*
 mplate_form_element attribute), 170
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_v-*
 ersion_form_element attribute), 171
`group` (*fobi.contrib.apps.drf_integration.form_elements.content.content_v-*
 ersion_form_element attribute), 171
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.boolean.bas-*
 ic_form_element attribute), 172
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.boolean.fobi_in-*
 tegration_form_element attribute), 172
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_se-*
 lection_form_element attribute), 173
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_se-*
 lection_form_element attribute), 174
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.date.base.D-*
 ateFormElement attribute), 175
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.date.fobi_in-*
 tegration_form_element attribute), 175
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_d-*
 own_form_element attribute), 176
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_d-*
 own_form_element attribute), 176
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.datetime.bas-*
 icFormElement attribute), 177
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.datetime.fobi_in-*
 tegration_form_element attribute), 177
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.decimal.bas-*
 icFormElement attribute), 178
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.decimal.fobi_in-*
 tegration_form_element attribute), 178
`group` (*fobi.contrib.apps.drf_integration.form_elements.fields.duration.bas-*
 icFormElement attribute), 178

[illegible]

| | |
|--------------|------------|
| Index | 571 |
|--------------|------------|

[attribute](#)), 288
[group \(fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.fobi_form_elements.SelectMultipleMPTTModel](#)
[attribute](#)), 289
[group \(fobi.contrib.plugins.form_elements.fields.select_multiple_with_max_input_plugin.fobi_form_elements.SelectMultipleWithMaxInputPlugin](#)
[attribute](#)), 290
[group \(fobi.contrib.plugins.form_elements.fields.select_multiple_with_max_input_plugin.fobi_form_elements.SelectMultipleWithMaxInputPlugin](#)
[attribute](#)), 292
[group \(fobi.contrib.plugins.form_elements.fields.slider.base.SliderInputPlugin](#)
[attribute](#)), 294
[group \(fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements.SliderInputPlugin](#) (in module
[attribute](#)), 295
[group \(fobi.contrib.plugins.form_elements.fields.slug.base.SlugInputPlugin](#)
[attribute](#)), 297
[group \(fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin](#)
[attribute](#)), 297
[group \(fobi.contrib.plugins.form_elements.fields.text.base.TextInputPlugin](#) (in module
[attribute](#)), 299
[group \(fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin](#)
[attribute](#)), 299
[group \(fobi.contrib.plugins.form_elements.fields.textarea.base.TextareaPlugin](#)
[attribute](#)), 300
[group \(fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements.TextareaPlugin](#)
[attribute](#)), 301
[group \(fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin](#)
[attribute](#)), 302
[group \(fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin](#)
[attribute](#)), 302
[group \(fobi.contrib.plugins.form_elements.fields.url.base.URLInputPlugin](#)
[attribute](#)), 304
[group \(fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin](#)
[attribute](#)), 304
[group \(fobi.contrib.plugins.form_elements.security.captcha.base.CaptchaInputPlugin](#)
[attribute](#)), 306
[group \(fobi.contrib.plugins.form_elements.security.captcha.fobi_form_elements.CaptchaInputPlugin](#)
[attribute](#)), 306
[group \(fobi.contrib.plugins.form_elements.security.honeypot.base.HoneypotInputPlugin](#)
[attribute](#)), 308
[group \(fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_elements.HoneypotInputPlugin](#)
[attribute](#)), 309
[group \(fobi.contrib.plugins.form_elements.security.recaptcha.base.ReCaptchaInputPlugin](#) (class in
[attribute](#)), 310
[group \(fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements.ReCaptchaInputPlugin](#)
[attribute](#)), 310
[group \(fobi.contrib.plugins.form_elements.test.dummy.base.DummyPlugin](#) (class in
[attribute](#)), 312
[group \(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyPlugin](#)
[attribute](#)), 312
[groups \(fobi.models.AbstractPluginModel attribute\),](#) [hide_success_page_title](#)
[415](#)
[groups \(fobi.models.FormElement attribute\),](#) [416](#)
[groups \(fobi.models.FormHandler attribute\),](#) [417](#)
[groups \(fobi.models.FormWizardHandler attribute\),](#) [418](#)
[groups_list \(\)](#) (fobi.models.AbstractPluginModel [HoneypotInputForm](#) (class in

H

| | | |
|---|---|---|
| <code>fobi.contrib.plugins.form_elements.security.honeypot</code> | <code>(fobi.models.FormEntry attribute)</code> | 423 |
| <code>309</code> | <code>id (fobi.models.FormFieldsetEntry attribute)</code> | 424 |
| <code>HoneypotInputPlugin</code> | <code>(class in id (fobi.models.FormHandler attribute)</code> | 417 |
| <code>fobi.contrib.plugins.form_elements.security.honeypot</code> | <code>(fobi.models.FormHandlerEntry attribute)</code> | 425 |
| <code>307</code> | <code>id (fobi.models.FormWizardEntry attribute)</code> | 426 |
| <code>HoneypotInputPlugin</code> | <code>(class in id (fobi.models.FormWizardFormEntry attribute)</code> | 428 |
| <code>fobi.contrib.plugins.form_elements.security.honeypot</code> | <code>(fobi.models.FormWizardHandler attribute)</code> | 418 |
| <code>309</code> | <code>id (fobi.models.FormWizardHandlerEntry attribute)</code> | 429 |
| <code>html_class (fobi.base.BasePlugin attribute)</code> | <code>381</code> | <code>import_data() (fobi.form_importers.BaseFormImporter</code> |
| <code>html_classes (fobi.base.BasePlugin attribute)</code> | <code>381</code> | <code>method)</code> |
| <code>html_classes (fobi.contrib.plugins.form_elements.content.content_markdown_base)</code> | <code>ContentMarkdownPlugin</code> | 437 |
| <code>attribute)</code> | <code>226</code> | <code>import_form_entry_ajax_template</code> |
| <code>html_classes (fobi.contrib.plugins.form_elements.content.content_markdown_widgets.BaseContentMarkdownPluginWidget</code> | <code>(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> | |
| <code>attribute)</code> | <code>228</code> | <code>358</code> |
| <code>html_classes (fobi.contrib.plugins.form_elements.content.content_rich_text_widgets.BaseContentRichTextPluginWidget</code> | <code>attribute)</code> | 231 |
| <code>attribute)</code> | <code>231</code> | <code>import_form_entry_ajax_template</code> |
| <code>html_classes (fobi.contrib.plugins.form_elements.fields.slider_base)</code> | <code>SliderPlugin</code> | 354 |
| <code>attribute)</code> | <code>294</code> | <code>attribute)</code> |
| <code>html_classes (fobi.contrib.plugins.form_elements.fields.slider_widgets.BaseSliderPluginWidget</code> | <code>(fobi.contrib.themes.djangoCMS.admin_style_theme.fobi_themes.</code> | |
| <code>attribute)</code> | <code>295</code> | <code>344</code> |
| <code>html_classes (fobi.contrib.plugins.form_elements.fields.slider_widgets.BaseSliderPluginWidget</code> | <code>attribute)</code> | 296 |
| <code>attribute)</code> | <code>296</code> | <code>import_form_entry_template</code> |
| <code>html_id (fobi.base.BasePlugin attribute)</code> | <code>381</code> | <code>(fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> |
| <code>HTTPRepostForm</code> | <code>(class in id (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme</code> | |
| <code>fobi.contrib.plugins.form_handlers.http_repost.form</code> | <code>import_form_entry_template</code> | 350 |
| <code>325</code> | <code>(fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> | |
| <code>HTTPRepostHandlerPlugin</code> | <code>(class in id (fobi.contrib.themes.simple.fobi_themes.SimpleTheme</code> | |
| <code>fobi.contrib.apps.drf_integration.form_handlers.http_repost_base)</code> | <code>http_repost_wizard_entry()</code> | 437 |
| <code>208</code> | <code>fobi.views)</code> | 437 |
| <code>HTTPRepostHandlerPlugin</code> | <code>(class in ImportFormEntryForm (class in fobi.forms)</code> | 408 |
| <code>fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handlers)</code> | <code>(class in</code> | |
| <code>208</code> | <code>fobi.forms)</code> | 408 |
| <code>HTTPRepostHandlerPlugin</code> | <code>(class in ImproperlyConfigured)</code> | 402 |
| <code>fobi.contrib.plugins.form_handlers.http_repost_base)</code> | <code>active_page_message (fobi.models.FormEntry</code> | |
| <code>323</code> | <code>attribute)</code> | 423 |
| <code>HTTPRepostHandlerPlugin</code> | <code>(class in inactive_page_title (fobi.models.FormEntry at-</code> | |
| <code>fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers)</code> | <code>423</code> | |
| <code>324</code> | <code>initial (fobi.contrib.apps.drf_integration.fields.NoneField</code> | |
| <code>HTTPRepostWizardHandlerPlugin</code> | <code>(class in id (fobi.contrib.apps.drf_integration.fields.NoneField</code> | |
| <code>fobi.contrib.plugins.form_handlers.http_repost_base)</code> | <code>initial_dict (fobi.wizard.views.dynamic.DynamicWizardView</code> | |
| <code>323</code> | <code>attribute)</code> | 370 |
| <code>HTTPRepostWizardHandlerPlugin</code> | <code>(class in inlines (fobi.admin.FormEntryAdmin attribute)</code> | 375 |
| <code>fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers)</code> | <code>(fobi.admin.FormWizardEntryAdmin at-</code> | |
| <code>324</code> | <code>tribute)</code> | 376 |
| <code>input_type (fobi.widgets.NumberInput attribute)</code> | <code>438</code> | |
| <code>InputForm</code> | <code>(class in</code> | |
| <code>id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry)</code> | <code>(fobi.contrib.plugins.form_elements.fields.input.forms)</code> | |
| <code>attribute)</code> | <code>320</code> | 261 |
| <code>id (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormEntry)</code> | <code>(class in</code> | |
| <code>attribute)</code> | <code>321</code> | <code>fobi.contrib.apps.drf_integration.form_elements.fields.input.base)</code> |
| <code>id (fobi.models.FormElement attribute)</code> | <code>416</code> | 185 |
| <code>id (fobi.models.FormElementEntry attribute)</code> | <code>421</code> | <code>InputPlugin</code> |
| | | <code>(class in</code> |

[fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration_form_elements\),](#)
[185](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.con](#)
[InputPlugin](#) (class in [attribute\), 171](#)
[fobi.contrib.plugins.form_elements.fields.input.base\),](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.con](#)
[260](#) [attribute\), 171](#)
[InputPlugin](#) (class in [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[fobi.contrib.plugins.form_elements.fields.input.fobi_form_elements\),](#) [172](#)
[260](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[insert\(\) \(fobi.data_structures.SortableDict method\),](#) [attribute\), 173](#)
[398](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[insert_after_key\(\)](#) [attribute\), 173](#)
[\(fobi.data_structures.SortableDict method\),](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[398](#) [attribute\), 174](#)
[insert_before_key\(\)](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[\(fobi.data_structures.SortableDict method\),](#) [attribute\), 175](#)
[398](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[instance_dict\(fobi.wizard.views.dynamic.DynamicWizardView](#) [attribute\), 175](#)
[attribute\), 370](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[IntegerInputForm](#) (class in [attribute\), 176](#)
[fobi.contrib.plugins.form_elements.fields.integer.forms\),](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[263](#) [attribute\), 176](#)
[IntegerInputPlugin](#) (class in [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[fobi.contrib.apps.drf_integration.form_elements.fields.integer.attribute\),](#) [177](#)
[186](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[IntegerInputPlugin](#) (class in [attribute\), 177](#)
[fobi.contrib.apps.drf_integration.form_elements.fields.integer.fobi_integration_form_elements\),](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[186](#) [attribute\), 178](#)
[IntegerInputPlugin](#) (class in [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[fobi.contrib.plugins.form_elements.fields.integer.base\),](#) [attribute\), 178](#)
[262](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[IntegerInputPlugin](#) (class in [attribute\), 179](#)
[fobi.contrib.plugins.form_elements.fields.integer.fobi_form_elements\),](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[262](#) [attribute\), 179](#)
[integrate_with\(fobi.base.IntegrationFormCallback](#) [integrate_with\(fobi.contrib.apps.drf_integration.form_elements.field](#)
[attribute\), 394](#) [attribute\), 180](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 166](#) [attribute\), 181](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 166](#) [attribute\), 181](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 167](#) [attribute\), 182](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 167](#) [attribute\), 183](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 168](#) [attribute\), 183](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 168](#) [attribute\), 184](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 169](#) [attribute\), 184](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 169](#) [attribute\), 185](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 170](#) [attribute\), 185](#)
[integrate_with\(fobi.contrib.apps.drf_integration.form_elements.content_to\(fobi.contrib.apps.drf_integration.plugin_elements.field](#)
[attribute\), 170](#) [attribute\), 185](#)

[illegible]

IntegrationFormHandlerPluginRegistry (class in *fobi.base*), 395

IntegrationProcessor (class in *fobi.integration.processors*), 355

InvalidRegistryItemType, 402

IPAddressInputForm (class in *fobi.contrib.plugins.form_elements.fields.ip_address.forms*), 264

IPAddressInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base*), 187

IPAddressInputPlugin (class in *fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration_form_elements*), 187

IPAddressInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.ip_address.base*), 263

IPAddressInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.ip_address.fobi_integration_form_elements*), 264

is_active (*fobi.models.FormEntry* attribute), 423

is_ajax() (*fobi.helpers.StrippedRequest* method), 413

is_anonymous() (*fobi.helpers.StrippedUser* method), 413

is_app_setup_completed() (in module *fobi.tests.core*), 363

is_cloneable (*fobi.models.FormEntry* attribute), 423

is_cloneable (*fobi.models.FormWizardEntry* attribute), 426

is_hidden (*fobi.base.FormElementPlugin* attribute), 386

is_hidden (*fobi.base.IntegrationFormElementPlugin* attribute), 395

is_hidden (*fobi.base.IntegrationFormHandlerPlugin* attribute), 395

is_hidden (*fobi.contrib.plugins.form_elements.fields.hidden.base.HiddenInputPlugin* attribute), 259

is_hidden (*fobi.contrib.plugins.form_elements.fields.hidden.fobi_integration_form_elements.HiddenInputPlugin* attribute), 259

is_hidden (*fobi.contrib.plugins.form_elements.security.honeypot.base.HoneypotInputPlugin* attribute), 308

is_hidden (*fobi.contrib.plugins.form_elements.security.honeypot.fobi_integration_form_elements.HoneypotInputPlugin* attribute), 309

is_hidden (*fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget* attribute), 331

is_public (*fobi.models.FormEntry* attribute), 423

is_public (*fobi.models.FormWizardEntry* attribute), 426

is_repeatable (*fobi.models.FormFieldsetEntry* attribute), 424

is_secure() (*fobi.helpers.StrippedRequest* method), 413

items() (*fobi.base.BaseRegistry* method), 384

items() (*fobi.data_structures.SortableDict* method), 398

iterable_to_dict() (in module *fobi.helpers*), 412

J

js (*fobi.reusable.markdown_widget.widgets.MarkdownWidget.Media* attribute), 360

JSONDataExporter (class in *fobi.helpers*), 412

K

keys() (*fobi.data_structures.SortableDict* method), 398

L

label (*fobi.apps.Config* attribute), 378

label (*fobi.base.BaseFormFieldPluginForm* attribute), 378

label (*fobi.contrib.apps.djangocms_integration.apps.Config* attribute), 164

label (*fobi.contrib.apps.drf_integration.apps.Config* attribute), 210

label (*fobi.contrib.apps.drf_integration.form_elements.content.content_in* attribute), 165

label (*fobi.contrib.apps.drf_integration.form_elements.content.content_in* attribute), 166

label (*fobi.contrib.apps.drf_integration.form_elements.content.content_m* attribute), 167

label (*fobi.contrib.apps.drf_integration.form_elements.content.content_r* attribute), 169

label (*fobi.contrib.apps.drf_integration.form_elements.content.content_t* attribute), 170

label (*fobi.contrib.apps.drf_integration.form_elements.content.content_v* attribute), 171

label (*fobi.contrib.apps.drf_integration.form_elements.fields.boolean.app* attribute), 172

label (*fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_se* attribute), 173

label (*fobi.contrib.apps.drf_integration.form_elements.fields.date.apps.C* attribute), 174

label (*fobi.contrib.apps.drf_integration.form_elements.fields.date_drop_c* attribute), 175

label (*fobi.contrib.apps.drf_integration.form_elements.fields.datetime.app* attribute), 176

label (*fobi.contrib.apps.drf_integration.form_elements.fields.decimal.app* attribute), 177

label (*fobi.contrib.apps.drf_integration.form_elements.fields.duration.app* attribute), 178

label (*fobi.contrib.apps.drf_integration.form_elements.fields.email.apps.C* attribute), 180

label (*fobi.contrib.apps.drf_integration.form_elements.fields.file.apps.Co* attribute), 181

label (*fobi.contrib.apps.drf_integration.form_elements.fields.float.apps.C* attribute), 183

label (*fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps* attribute), 184

label (fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi.contrib.plugins.form_elements.content.content_richtext.apps.Config attribute), 185

label (fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi.contrib.plugins.form_elements.content.content_text.apps.Config attribute), 186

label (fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi.contrib.plugins.form_elements.content.content_video.apps.Config attribute), 187

label (fobi.contrib.apps.drf_integration.form_elements.fields.null.fobi.contrib.plugins.form_elements.fields.boolean.apps.Config attribute), 188

label (fobi.contrib.apps.drf_integration.form_elements.fields.pass.fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps.Config attribute), 189

label (fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi.contrib.plugins.form_elements.fields.date.apps.Config attribute), 190

label (fobi.contrib.apps.drf_integration.form_elements.fields.range.fobi.contrib.plugins.form_elements.fields.date_drop_down.apps.Config attribute), 191

label (fobi.contrib.apps.drf_integration.form_elements.fields.range.fobi.contrib.plugins.form_elements.fields.datetime.apps.Config attribute), 192

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.decimal.apps.Config attribute), 193

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.duration.apps.Config attribute), 194

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobimultipleplugins.fobi.contrib.plugins.form_elements.fields.email.apps.Config attribute), 195

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobimultipleplugins.fobi.contrib.plugins.form_elements.fields.file.apps.Config attribute), 197

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobimultipleplugins.fobi.contrib.plugins.form_elements.fields.float.apps.Config attribute), 199

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.hidden.apps.Config attribute), 200

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.input.apps.Config attribute), 201

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.integer.apps.Config attribute), 202

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config attribute), 203

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config attribute), 204

label (fobi.contrib.apps.drf_integration.form_elements.fields.select.fobi.contrib.plugins.form_elements.fields.password.apps.Config attribute), 205

label (fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi.contrib.plugins.form_elements.fields.radio.apps.Config attribute), 207

label (fobi.contrib.apps.drf_integration.form_handlers.http_post.fobi.contrib.plugins.form_elements.fields.range_select.apps.Config attribute), 208

label (fobi.contrib.apps.drf_integration.form_handlers.mailchimp.fobi.contrib.plugins.form_elements.fields.regex.apps.Config attribute), 209

label (fobi.contrib.apps.feincms_integration.apps.Config label (fobi.contrib.plugins.form_elements.fields.select.apps.Config attribute), 216

label (fobi.contrib.apps.mezzanine_integration.apps.Config label (fobi.contrib.plugins.form_elements.fields.select_model_object.apps.Config attribute), 218

label (fobi.contrib.plugins.form_elements.content.content_image.fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps.Config attribute), 220

label (fobi.contrib.plugins.form_elements.content.content_image.fobi.contrib.plugins.form_elements.fields.select_multiple.apps.Config attribute), 223

label (fobi.contrib.plugins.form_elements.content.content_image.fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.apps.Config attribute), 226

- label (fobi.contrib.plugins.form_elements.fields.select_multiple_in_foundation5.widgets.form_elements.content_rich_text attribute), 288
- label (fobi.contrib.plugins.form_elements.fields.select_multiple_in_foundation5.widgets.form_elements.date_foundation5 attribute), 290
- label (fobi.contrib.plugins.form_elements.fields.slider.apps.Config (fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5 attribute), 293
- label (fobi.contrib.plugins.form_elements.fields.slug.apps.Config (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5 attribute), 296
- label (fobi.contrib.plugins.form_elements.fields.text.apps.Config (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5 attribute), 298
- label (fobi.contrib.plugins.form_elements.fields.textarea.apps.Config (fobi.contrib.themes.simple.apps.Config attribute), 300
- label (fobi.contrib.plugins.form_elements.fields.time.apps.Config (fobi.contrib.themes.simple.widgets.form_elements.content_richtext attribute), 301
- label (fobi.contrib.plugins.form_elements.fields.url.apps.Config (fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps.Config attribute), 303
- label (fobi.contrib.plugins.form_elements.security.captcha.apps.Config (fobi.contrib.apps.drf_integration.metadata.FobiMetadata attribute), 305
- label (fobi.contrib.plugins.form_elements.security.honeypot.apps.Config (fobi.admin.BasePluginModelAdmin attribute), 307
- label (fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config (fobi.admin.FormElementEntryAdmin attribute), 310
- label (fobi.contrib.plugins.form_elements.test.dummy.apps.Config (fobi.admin.FormEntryAdmin attribute), 311
- label (fobi.contrib.plugins.form_handlers.db_store.apps.Config (fobi.admin.ModelFormEntryAdmin attribute), 315
- label (fobi.contrib.plugins.form_handlers.http_repost.apps.Config (fobi.admin.FormHandlerEntryAdmin attribute), 323
- label (fobi.contrib.plugins.form_handlers.mail.apps.Config (fobi.admin.FormWizardEntryAdmin attribute), 326
- label (fobi.contrib.plugins.form_importers.mailchimp_importer.apps.Config (fobi.contrib.plugins.form_handlers.db_store.admin.SavedForm attribute), 332
- label (fobi.contrib.themes.bootstrap3.apps.Config attribute), 339
- label (fobi.contrib.themes.bootstrap3.widgets.form_elements.content_rich_text_bootstrap3 (fobi.admin.ModelFormEntryAdmin attribute), 335
- label (fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext_bootstrap3 (fobi.admin.ModelFormEntryAdmin attribute), 336
- label (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3 (fobi.admin.ModelFormEntryAdmin attribute), 336
- label (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3 (fobi.admin.ModelFormEntryAdmin attribute), 337
- label (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3 (fobi.admin.ModelFormEntryAdmin attribute), 338
- label (fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3 (fobi.admin.ModelFormEntryAdmin attribute), 338
- label (fobi.contrib.themes.djangocms_admin_style_theme.apps.Config (fobi.admin.ModelFormEntryAdmin attribute), 343
- label (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.content_richtext_djangocms_admin_style_theme (fobi.admin.ModelFormEntryAdmin attribute), 341
- label (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements.date_djangocms_admin_style_theme (fobi.admin.ModelFormEntryAdmin attribute), 342
- label (fobi.contrib.themes.foundation5.apps.Config attribute), 349

list_filter (fobi.contrib.plugins.form_handlers.db_storage_admin.SavedFormWizardDataEntryAdmin module attribute), 315
 lists_overlap() (in module fobi.helpers), 412
 LIVE_SERVER_URL (fobi.tests.base.BaseFobiBrowserBuildDynamicFormTest.to_label() (in module attribute), 362
 load_plugin_data() (fobi.base.BasePlugin mark_app_setup_as_completed() (in module method), 381
 login_required_template_name MarkdownWidget (class in (fobi.integration.processors.IntegrationProcessor fobi.reusable.markdown_widget.widgets), attribute), 355
 lookup_field (fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet Media (class in attribute), 215
 lookup_url_kwarg (fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet Media (class in attribute), 215
 master_base_template (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 340
M
 MailchimpAPIKeyForm (class in master_base_template (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes. attribute), 344
 333
 MailChimpImporter (class in master_base_template (fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme attribute), 350
 332
 MailChimpImporter (class in master_base_template (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 354
 333
 MailchimpImporterWizardView (class in media (fobi.admin.BasePluginModelAdmin attribute), 374
 334
 MailchimpListIDForm (class in media (fobi.admin.FormElementEntryAdmin attribute), 374
 334
 MailForm (class in fobi.contrib.plugins.form_handlers.mail.forms), 375
 330
 MailHandlerMixin (class in media (fobi.admin.FormFieldsetEntryAdmin attribute), 375
 331
 MailHandlerPlugin (class in media (fobi.admin.FormHandlerAdmin attribute), 376
 209
 MailHandlerPlugin (class in media (fobi.admin.FormHandlerEntryAdmin attribute), 376
 209
 MailHandlerPlugin (class in media (fobi.admin.FormHandlerEntryInlineAdmin attribute), 376
 209
 MailHandlerPlugin (class in media (fobi.admin.FormWizardFormEntryInlineAdmin attribute), 377
 326
 MailHandlerPlugin (class in media (fobi.admin.FormWizardHandlerAdmin attribute), 377
 329
 MailWizardHandlerPlugin (class in media (fobi.admin.FormWizardHandlerEntryInlineAdmin attribute), 377
 326
 MailWizardHandlerPlugin (class in media (fobi.contrib.plugins.form_elements.content.content_image.forms.C attribute), 222
 329
 MailWizardHandlerPlugin (class in media (fobi.contrib.plugins.form_elements.content.content_image_url.form attribute), 225
 329
 MailWizardHandlerPlugin (class in media (fobi.contrib.plugins.form_elements.content.content_markdown.form attribute), 228
 329

media (fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm attribute), 231

media (fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm attribute), 234

media (fobi.contrib.plugins.form_elements.content.content_image.forms.ContentImageForm attribute), 236

media (fobi.contrib.plugins.form_elements.fields.boolean.forms.BooleanField attribute), 238

media (fobi.contrib.plugins.form_elements.fields.checkbox.forms.BooleanField attribute), 241

media (fobi.contrib.plugins.form_elements.fields.date.forms.DateField attribute), 243

media (fobi.contrib.plugins.form_elements.fields.date_dropdown.forms.DateField attribute), 245

media (fobi.contrib.plugins.form_elements.fields.datetime.forms.DateTimeField attribute), 247

media (fobi.contrib.plugins.form_elements.fields.decimal.forms.DecimalField attribute), 249

media (fobi.contrib.plugins.form_elements.fields.duration.forms.DurationField attribute), 251

media (fobi.contrib.plugins.form_elements.fields.email.forms.EmailForm attribute), 253

media (fobi.contrib.plugins.form_elements.fields.file.forms.FileField attribute), 256

media (fobi.contrib.plugins.form_elements.fields.float.forms.FloatField attribute), 258

media (fobi.contrib.plugins.form_elements.fields.hidden.forms.HiddenForm attribute), 260

media (fobi.contrib.plugins.form_elements.fields.input.forms.TextInputForm attribute), 261

media (fobi.contrib.plugins.form_elements.fields.integer.forms.IntegerField attribute), 263

media (fobi.contrib.plugins.form_elements.fields.ip_address.forms.IPAddressForm attribute), 265

media (fobi.contrib.plugins.form_elements.fields.null_boolean.forms.NullBooleanField attribute), 266

media (fobi.contrib.plugins.form_elements.fields.password.forms.PasswordForm attribute), 268

media (fobi.contrib.plugins.form_elements.fields.radio.forms.BooleanField attribute), 270

media (fobi.contrib.plugins.form_elements.fields.range_select.forms.BooleanField attribute), 273

media (fobi.contrib.plugins.form_elements.fields.regex.forms.RegexForm attribute), 275

media (fobi.contrib.plugins.form_elements.fields.select.forms.SelectForm attribute), 277

media (fobi.contrib.plugins.form_elements.fields.select_model_object.forms.SelectModelObjectInputForm attribute), 280

media (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.forms.SelectMPTTModelObjectInputForm attribute), 282

media (fobi.contrib.plugins.form_elements.fields.select_multiple.forms.SelectMultipleForm attribute), 285

media (fobi.contrib.plugins.form_elements.fields.select_multiple_and_model_objects.forms.SelectMultipleModelObjectInputForm attribute), 287

media (fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms.SelectMultipleMPTTModelObjectInputForm attribute), 290

media (fobi.contrib.plugins.form_elements.fields.select_multiple_with_model_objects.forms.SelectMultipleWithModelObjectInputForm attribute), 293

media (fobi.contrib.plugins.form_elements.fields.slider.forms.SliderInputForm attribute), 296

media (fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm attribute), 298

media (fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm attribute), 300

media (fobi.contrib.plugins.form_elements.fields.textarea.forms.TextareaForm attribute), 301

media (fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm attribute), 303

media (fobi.contrib.plugins.form_elements.fields.url.forms.URLInputForm attribute), 305

media (fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaForm attribute), 307

media (fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneyPotForm attribute), 309

media (fobi.contrib.plugins.form_elements.security.recaptcha.forms.ReCaptchaForm attribute), 311

media (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormData attribute), 314

media (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizard attribute), 315

media (fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm attribute), 326

media (fobi.contrib.plugins.form_handlers.mail.forms.MailForm attribute), 330

media (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget attribute), 331

media (fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailChimpImporterForm attribute), 333

media (fobi.contrib.plugins.form_importers.mailchimp_importer.forms.MailChimpImporterForm attribute), 334

media (fobi.contrib.plugins.form_plugins.bulk_change_form_element_plugins_form attribute), 404

media (fobi.contrib.plugins.form_plugins.bulk_change_form_handler_plugins_form attribute), 405

media (fobi.contrib.plugins.form_plugins.bulk_change_form_wizard_handler_plugins_form attribute), 405

media (fobi.contrib.plugins.form_plugins.form_element_entry_form attribute), 405

media (fobi.contrib.plugins.form_plugins.form_entry_form attribute), 406

media (fobi.contrib.plugins.form_plugins.form_fieldset_entry_form attribute), 406

media (fobi.contrib.plugins.form_plugins.form_handler_entry_form attribute), 406

media (fobi.contrib.plugins.form_plugins.form_handler_form attribute), 407

media (fobi.contrib.plugins.form_plugins.form_plugin_widgets_form attribute), 407

media (fobi.contrib.plugins.form_plugins.form_size_form attribute), 407

media (fobi.contrib.plugins.form_plugins.form_size_form attribute), 408

[illegible]

| | | | | | |
|--------------------------|---|----|---|--------|----|
| Migration | (class | in | MultiEmailField | (class | in |
| | <i>fobi.migrations.0010_formwizardhandler</i>), | | <i>fobi.contrib.plugins.form_handlers.mail.fields</i>), | | |
| | 358 | | 328 | | |
| Migration | (class | in | MultiEmailWidget | (class | in |
| | <i>fobi.migrations.0011_formentry_title</i>), | | <i>fobi.contrib.plugins.form_handlers.mail.widgets</i>), | | |
| | 359 | | 331 | | |
| Migration | (class | in | MultipleChoiceWithMaxField | (class | in |
| | <i>fobi.migrations.0012_auto_20161109_1550</i>), | | <i>fobi.contrib.apps.drf_integration.fields</i>), | | |
| | 359 | | 213 | | |
| Migration | (class | in | MultipleChoiceWithMaxField | (class | in |
| | <i>fobi.migrations.0013_formwizardentry_show_all_navigation_buttons</i>), | | <i>fobi.contrib.plugins.form_elements.fields.select_multiple_with_max</i> | | |
| | 359 | | 292 | | |
| Migration | (class | in | N | | |
| | <i>fobi.migrations.0014_auto_20170516_1413</i>), | | name (<i>fobi.apps.Config</i> attribute), | | |
| | 359 | | 378 | | |
| model | (<i>fobi.admin.FormElementEntryInlineAdmin</i> at- | | name (<i>fobi.base.BaseFormFieldPluginForm</i> attribute), | | |
| | tribute), | | 378 | | |
| model | (<i>fobi.admin.FormHandlerEntryInlineAdmin</i> at- | | name (<i>fobi.base.BasePlugin</i> attribute), | | |
| | tribute), | | 381 | | |
| model | (<i>fobi.admin.FormWizardFormEntryInlineAdmin</i> | | name (<i>fobi.contrib.apps.djangocms_integration.apps.Config</i> | | |
| | attribute), | | attribute), | | |
| | 377 | | 164 | | |
| model | (<i>fobi.admin.FormWizardHandlerEntryInlineAdmin</i> | | name (<i>fobi.contrib.apps.drf_integration.apps.Config</i> at- | | |
| | attribute), | | tribute), | | |
| | 377 | | 210 | | |
| model | (<i>fobi.contrib.apps.drf_integration.serializers.FormEntrySerializer.Meta</i> | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_in</i> | | |
| | attribute), | | heritance), | | |
| | 214 | | 165 | | |
| model | (<i>fobi.forms.BulkChangeFormElementPluginsForm.Meta</i> | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_in</i> | | |
| | attribute), | | heritance), | | |
| | 404 | | 166 | | |
| model | (<i>fobi.forms.BulkChangeFormHandlerPluginsForm.Meta</i> | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_in</i> | | |
| | attribute), | | heritance), | | |
| | 404 | | 166 | | |
| model | (<i>fobi.forms.BulkChangeFormWizardHandlerPluginsForm.Meta</i> | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_in</i> | | |
| | attribute), | | heritance), | | |
| | 405 | | 166 | | |
| model | (<i>fobi.forms.FormElementEntryForm.Meta</i> at- | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_in</i> | | |
| | tribute), | | heritance), | | |
| | 405 | | 167 | | |
| model | (<i>fobi.forms.FormEntryForm.Meta</i> attribute), | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_in</i> | | |
| | 406 | | heritance), | | |
| model | (<i>fobi.forms.FormFieldsetEntryForm.Meta</i> at- | | tribute), | | |
| | tribute), | | 167 | | |
| | 406 | | 167 | | |
| model | (<i>fobi.forms.FormHandlerEntryForm.Meta</i> at- | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_ma</i> | | |
| | tribute), | | tribute), | | |
| | 406 | | 168 | | |
| model | (<i>fobi.forms.FormHandlerForm.Meta</i> attribute), | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_ma</i> | | |
| | 407 | | tribute), | | |
| | | | 168 | | |
| model | (<i>fobi.forms.FormWizardEntryForm.Meta</i> at- | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_ric</i> | | |
| | tribute), | | tribute), | | |
| | 407 | | 169 | | |
| model | (<i>fobi.forms.FormWizardFormEntryForm.Meta</i> at- | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_ric</i> | | |
| | tribute), | | tribute), | | |
| | 407 | | 169 | | |
| model | (<i>fobi.forms.FormWizardHandlerEntryForm.Meta</i> | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_ric</i> | | |
| | attribute), | | tribute), | | |
| | 408 | | 169 | | |
| ModelChoiceField | (class | in | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_tex</i> | | |
| | <i>fobi.contrib.apps.drf_integration.fields</i>), | | tribute), | | |
| | 213 | | 170 | | |
| ModelMultipleChoiceField | (class | in | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_tex</i> | | |
| | <i>fobi.contrib.apps.drf_integration.fields</i>), | | tribute), | | |
| | 213 | | 170 | | |
| move_after_key() | (<i>fobi.data_structures.SortableDict</i> | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_tex</i> | | |
| | method), | | tribute), | | |
| | 398 | | 170 | | |
| move_before_key() | (<i>fobi.data_structures.SortableDict</i> | | name (<i>fobi.contrib.apps.drf_integration.form_elements.content.content_vid</i> | | |
| | method), | | tribute), | | |
| | 399 | | 171 | | |

name (fobi.contrib.apps.drf_integration.form_elements.container(fobi.contrib.apps.drf_integration.form_elements.fields.file.base.FileField attribute), 171 attribute), 181

name (fobi.contrib.apps.drf_integration.form_elements.container(fobi.contrib.apps.drf_integration.form_elements.fields.file.fobi_integration attribute), 171 attribute), 182

name (fobi.contrib.apps.drf_integration.form_elements.fields.base(fobi.contrib.apps.drf_integration.form_elements.fields.float.apps.Config attribute), 172 attribute), 183

name (fobi.contrib.apps.drf_integration.form_elements.fields.base(fobi.contrib.apps.drf_integration.form_elements.fields.float.base.FloatField attribute), 172 attribute), 183

name (fobi.contrib.apps.drf_integration.form_elements.fields.base(fobi.contrib.apps.drf_integration.form_elements.fields.boolean.apps.Config attribute), 173 attribute), 183

name (fobi.contrib.apps.drf_integration.form_elements.fields.base(fobi.contrib.apps.drf_integration.form_elements.fields.hidden.apps.Config attribute), 173 attribute), 184

name (fobi.contrib.apps.drf_integration.form_elements.fields.base(fobi.contrib.apps.drf_integration.form_elements.fields.hidden.fobi_integration attribute), 173 attribute), 184

name (fobi.contrib.apps.drf_integration.form_elements.fields.base(fobi.contrib.apps.drf_integration.form_elements.fields.checkbox.apps.Config attribute), 174 attribute), 184

name (fobi.contrib.apps.drf_integration.form_elements.fields.base(fobi.contrib.apps.drf_integration.form_elements.fields.checkbox.fobi_integration attribute), 174 attribute), 184

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.input.apps.Config attribute), 175 attribute), 185

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.input.base.Input attribute), 175 attribute), 185

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.input.fobi_integration attribute), 175 attribute), 185

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.integer.apps.Config attribute), 176 attribute), 186

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.integer.base.IntegerField attribute), 176 attribute), 186

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.apps.Config attribute), 176 attribute), 186

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.base.IPAddressField attribute), 177 attribute), 187

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.ip_address.fobi_integration attribute), 177 attribute), 187

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.apps.Config attribute), 178 attribute), 188

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.base.BooleanField attribute), 178 attribute), 188

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.password.apps.Config attribute), 179 attribute), 189

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.password.base.PasswordField attribute), 179 attribute), 189

name (fobi.contrib.apps.drf_integration.form_elements.fields.date(fobi.contrib.apps.drf_integration.form_elements.fields.password.fobi_integration attribute), 179 attribute), 189

name (fobi.contrib.apps.drf_integration.form_elements.fields.email(fobi.contrib.apps.drf_integration.form_elements.fields.radio.apps.Config attribute), 180 attribute), 190

name (fobi.contrib.apps.drf_integration.form_elements.fields.email(fobi.contrib.apps.drf_integration.form_elements.fields.radio.base.RadioButton attribute), 180 attribute), 190

name (fobi.contrib.apps.drf_integration.form_elements.fields.email(fobi.contrib.apps.drf_integration.form_elements.fields.radio.fobi_integration attribute), 181 attribute), 190

name (fobi.contrib.apps.drf_integration.form_elements.fields.file(fobi.contrib.apps.drf_integration.form_elements.fields.range_select attribute), 181 attribute), 191

[illegible]

name (fobi.contrib.plugins.form_elements.content.content_image (fobi.contrib.plugins.form_elements.fields.url.plugin.fobi_form_element attribute), 225 attribute), 246

name (fobi.contrib.plugins.form_elements.content.content_marked (fobi.contrib.plugins.form_elements.fields.decimal.apps.Config attribute), 226 attribute), 248

name (fobi.contrib.plugins.form_elements.content.content_marked (fobi.contrib.plugins.form_elements.fields.decimal.base.DecimalInput attribute), 226 attribute), 248

name (fobi.contrib.plugins.form_elements.content.content_marked (fobi.contrib.plugins.form_elements.fields.duration.plugin.fobi_form_element attribute), 227 attribute), 248

name (fobi.contrib.plugins.form_elements.content.content_marked (fobi.contrib.plugins.form_elements.fields.duration.apps.Config attribute), 229 attribute), 249

name (fobi.contrib.plugins.form_elements.content.content_marked (fobi.contrib.plugins.form_elements.fields.duration.base.DurationInput attribute), 229 attribute), 250

name (fobi.contrib.plugins.form_elements.content.content_marked (fobi.contrib.plugins.form_elements.fields.duration.plugin.fobi_form_element attribute), 230 attribute), 250

name (fobi.contrib.plugins.form_elements.content.content_text (fobi.contrib.plugins.form_elements.fields.email.apps.Config attribute), 232 attribute), 252

name (fobi.contrib.plugins.form_elements.content.content_text (fobi.contrib.plugins.form_elements.fields.email.base.EmailInputPlugin attribute), 232 attribute), 252

name (fobi.contrib.plugins.form_elements.content.content_text (fobi.contrib.plugins.form_elements.fields.email.plugin.fobi_form_element attribute), 233 attribute), 252

name (fobi.contrib.plugins.form_elements.content.content_video (fobi.contrib.plugins.form_elements.fields.file.apps.Config attribute), 234 attribute), 253

name (fobi.contrib.plugins.form_elements.content.content_video (fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin attribute), 234 attribute), 254

name (fobi.contrib.plugins.form_elements.content.content_video (fobi.contrib.plugins.form_elements.fields.file.plugin.fobi_form_element attribute), 235 attribute), 255

name (fobi.contrib.plugins.form_elements.fields.boolean.apps.Config (fobi.contrib.plugins.form_elements.fields.float.apps.Config attribute), 237 attribute), 256

name (fobi.contrib.plugins.form_elements.fields.boolean.base.BooleanInputPlugin (fobi.contrib.plugins.form_elements.fields.float.base.FloatInputPlugin attribute), 237 attribute), 257

name (fobi.contrib.plugins.form_elements.fields.boolean.fobi_form_element (fobi.contrib.plugins.form_elements.fields.float.plugin.fobi_form_element attribute), 238 attribute), 257

name (fobi.contrib.plugins.form_elements.fields.checkbox_select (fobi.contrib.plugins.form_elements.fields.hidden.apps.Config attribute), 239 attribute), 258

name (fobi.contrib.plugins.form_elements.fields.checkbox_select (fobi.contrib.plugins.form_elements.fields.hidden.base.HiddenInputPlugin attribute), 239 attribute), 259

name (fobi.contrib.plugins.form_elements.fields.checkbox_select (fobi.contrib.plugins.form_elements.fields.hidden.plugin.fobi_form_element attribute), 240 attribute), 259

name (fobi.contrib.plugins.form_elements.fields.date.apps.Config (fobi.contrib.plugins.form_elements.fields.input.apps.Config attribute), 241 attribute), 260

name (fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin (fobi.contrib.plugins.form_elements.fields.input.base.InputPlugin attribute), 242 attribute), 260

name (fobi.contrib.plugins.form_elements.fields.date.fobi_form_element (fobi.contrib.plugins.form_elements.fields.input.plugin.fobi_form_element attribute), 242 attribute), 261

name (fobi.contrib.plugins.form_elements.fields.date_drop_down (fobi.contrib.plugins.form_elements.fields.integer.apps.Config attribute), 244 attribute), 262

name (fobi.contrib.plugins.form_elements.fields.date_drop_down (fobi.contrib.plugins.form_elements.fields.integer.base.IntegerInputPlugin attribute), 244 attribute), 262

name (fobi.contrib.plugins.form_elements.fields.date_drop_down (fobi.contrib.plugins.form_elements.fields.integer.plugin.fobi_form_element attribute), 244 attribute), 262

name (fobi.contrib.plugins.form_elements.fields.datetime.apps.Config (fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config attribute), 245 attribute), 263

name (fobi.contrib.plugins.form_elements.fields.datetime.base.DateTimeInputPlugin (fobi.contrib.plugins.form_elements.fields.ip_address.base.IPAddress attribute), 246 attribute), 264

name (fobi.contrib.plugins.form_elements.fields.url.fobi_form_handler.fobi_form_handlers.mail.fobi_form_handlers.MailWidget attribute), 304

name (fobi.contrib.plugins.form_elements.security.captcha.apps.Config (fobi.contrib.plugins.form_importers.mailchimp_importer.apps.Config attribute), 305

name (fobi.contrib.plugins.form_elements.security.captcha.base.CaptchaInputPlugin (fobi.contrib.plugins.form_importers.mailchimp_importer.base.MailCaptcha attribute), 306

name (fobi.contrib.plugins.form_elements.security.captcha.fobi_form_handler.fobi_form_handlers.mailchimp_importer.fobi_form_handlers attribute), 306

name (fobi.contrib.plugins.form_elements.security.honeypot.apps.Config (fobi.contrib.themes.bootstrap3.apps.Config attribute), 307

name (fobi.contrib.plugins.form_elements.security.honeypot.base.HoneypotInputPlugin (fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme attribute), 308

name (fobi.contrib.plugins.form_elements.security.honeypot.fobi_form_handler.fobi_form_handlers.mailchimp_importer.fobi_form_handlers attribute), 309

name (fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config (fobi.contrib.themes.bootstrap3.widgets.form_elements.content_richtext attribute), 310

name (fobi.contrib.plugins.form_elements.security.recaptcha.base.ReCaptchaInputPlugin (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap attribute), 310

name (fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_handler.fobi_form_handlers.mailchimp_importer.fobi_form_handlers attribute), 310

name (fobi.contrib.plugins.form_elements.test.dummy.apps.Config (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap attribute), 311

name (fobi.contrib.plugins.form_elements.test.dummy.base.DummyInputPlugin (fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap attribute), 312

name (fobi.contrib.plugins.form_elements.test.dummy.fobi_form_handler.fobi_form_handlers.djangocms_admin_style_theme.apps.Config attribute), 312

name (fobi.contrib.plugins.form_handlers.db_store.apps.Config (fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMS attribute), 315

name (fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_elements attribute), 315

name (fobi.contrib.plugins.form_handlers.db_store.base.DBStoreWidgetPlugin (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers attribute), 316

name (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handler.fobi_form_handlers.djangocms_admin_style_theme.fobi_themes.DjangoCMS attribute), 317

name (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handler.fobi_form_handlers.djangocms_admin_style_theme.fobi_themes.DjangoCMS attribute), 318

name (fobi.contrib.plugins.form_handlers.http_repost.apps.Config (fobi.contrib.themes.foundation5.widgets.form_elements.content_richtext attribute), 323

name (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostInputPlugin (fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation attribute), 323

name (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostWidgetPlugin (fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation attribute), 323

name (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handler.fobi_form_handlers.djangocms_admin_style_theme.fobi_themes.DjangoCMS attribute), 324

name (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handler.fobi_form_handlers.djangocms_admin_style_theme.fobi_themes.DjangoCMS attribute), 325

name (fobi.contrib.plugins.form_handlers.mail.apps.Config (fobi.contrib.themes.simple.apps.Config attribute), 326

name (fobi.contrib.plugins.form_handlers.mail.base.MailHandlerPlugin (fobi.contrib.themes.simple.fobi_themes.SimpleTheme attribute), 326

name (fobi.contrib.plugins.form_handlers.mail.base.MailWizardPlugin (fobi.contrib.themes.simple.widgets.form_elements.content_richtext attribute), 327

name (fobi.contrib.plugins.form_handlers.mail.fobi_form_handler.fobi_form_handlers.djangocms_admin_style_theme.fobi_themes.DjangoCMS attribute), 329

| name | value | name | value |
|---|-------|---|-------|
| <code>(fobi.form_importers.BaseFormImporter attribute), 403</code> | | <code>operations (fobi.migrations.0001_initial.Migration attribute), 357</code> | |
| <code>name (fobi.models.FormEntry attribute), 423</code> | | <code>operations (fobi.migrations.0002_auto_20150912_1744.Migration attribute), 357</code> | |
| <code>name (fobi.models.FormFieldsetEntry attribute), 425</code> | | <code>operations (fobi.migrations.0003_auto_20160517_1005.Migration attribute), 357</code> | |
| <code>name (fobi.models.FormWizardEntry attribute), 426</code> | | <code>operations (fobi.migrations.0004_auto_20160906_1513.Migration attribute), 357</code> | |
| <code>NoDefaultThemeSet, 402</code> | | <code>operations (fobi.migrations.0005_auto_20160908_1457.Migration attribute), 358</code> | |
| <code>NoneField (class in fobi.contrib.apps.drf_integration.fields), 213</code> | | <code>operations (fobi.migrations.0006_auto_20160911_1549.Migration attribute), 358</code> | |
| <code>NullBooleanSelectForm (class in fobi.contrib.plugins.form_elements.fields.null_boolean.forms), 266</code> | | <code>operations (fobi.migrations.0007_auto_20160926_1652.Migration attribute), 358</code> | |
| <code>NullBooleanSelectPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.plugins), 188</code> | | <code>operations (fobi.migrations.0008_formwizardhandlerentry.Migration attribute), 358</code> | |
| <code>NullBooleanSelectPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.null_boolean.plugins), 188</code> | | <code>operations (fobi.migrations.0009_formwizardentry_wizard_type.Migration attribute), 358</code> | |
| <code>NullBooleanSelectPlugin (class in fobi.contrib.plugins.form_elements.fields.null_boolean.plugins), 265</code> | | <code>operations (fobi.migrations.0010_formwizardhandler.Migration attribute), 358</code> | |
| <code>NullBooleanSelectPlugin (class in fobi.contrib.plugins.form_elements.fields.null_boolean.fobi.contrib.plugins), 265</code> | | <code>operations (fobi.migrations.0011_formentry_title.Migration attribute), 359</code> | |
| <code>NumberInput (class in fobi.widgets), 438</code> | | <code>operations (fobi.migrations.0012_auto_20161109_1550.Migration attribute), 359</code> | |
| O | | <code>operations (fobi.migrations.0013_formwizardentry_show_all_navigation.Migration attribute), 359</code> | |
| <code>objects (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormDefaultEntry attribute), 320</code> | | <code>operations (fobi.migrations.0014_auto_20170516_1413.Migration attribute), 359</code> | |
| <code>objects (fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardDataEntry attribute), 321</code> | | <code>options (fobi.contrib.plugins.fobi.contrib.plugins), 438</code> | |
| <code>objects (fobi.models.FormElement attribute), 416</code> | | <code>(fobi.widgets.RichSelectInverseQuotes attribute), 438</code> | |
| <code>objects (fobi.models.FormElementEntry attribute), 421</code> | | P | |
| <code>objects (fobi.models.FormEntry attribute), 423</code> | | <code>PasswordInputForm (class in fobi.contrib.plugins.form_elements.fields.password.forms), 268</code> | |
| <code>objects (fobi.models.FormFieldsetEntry attribute), 425</code> | | <code>PasswordInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.password.b), 189</code> | |
| <code>objects (fobi.models.FormHandler attribute), 417</code> | | <code>PasswordInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.password.f), 189</code> | |
| <code>objects (fobi.models.FormHandlerEntry attribute), 425</code> | | <code>PasswordInputPlugin (class in fobi.contrib.plugins.form_elements.fields.password.base), 267</code> | |
| <code>objects (fobi.models.FormWizardEntry attribute), 426</code> | | <code>PasswordInputPlugin (class in fobi.contrib.plugins.form_elements.fields.password.fobi_form_ele), 267</code> | |
| <code>objects (fobi.models.FormWizardFormEntry attribute), 428</code> | | <code>perform_form_entry_import() (in module fobi.contrib.plugins.drf_integration.views.fobi_form_entry_view_set), 215</code> | |
| <code>objects (fobi.models.FormWizardHandler attribute), 418</code> | | <code>perform_form_entry_import() (in module fobi.contrib.plugins.drf_integration.views.fobi_form_entry_view_set), 215</code> | |
| <code>objects (fobi.models.FormWizardHandlerEntry attribute), 429</code> | | <code>perform_form_entry_import() (in module fobi.contrib.plugins.drf_integration.views.fobi_form_entry_view_set), 215</code> | |
| <code>only_args (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormDefaultEntry attribute), 314</code> | | <code>perform_form_entry_import() (in module fobi.contrib.plugins.drf_integration.views.fobi_form_entry_view_set), 215</code> | |
| <code>only_args (fobi.contrib.plugins.form_handlers.db_store.admin.SavedFormWizardDataEntry attribute), 315</code> | | <code>perform_form_entry_import() (in module fobi.contrib.plugins.drf_integration.views.fobi_form_entry_view_set), 215</code> | |
| <code>operations (fobi.contrib.plugins.form_handlers.db_store.migrations.0001_initial.Migration attribute), 313</code> | | <code>perform_form_entry_import() (in module fobi.contrib.plugins.drf_integration.views.fobi_form_entry_view_set), 215</code> | |
| <code>operations (fobi.contrib.plugins.form_handlers.db_store.migrations.0002_auto_20150912_1744.Migration attribute), 313</code> | | <code>perform_form_entry_import() (in module fobi.contrib.plugins.drf_integration.views.fobi_form_entry_view_set), 215</code> | |

| | |
|------------------------|--|
| permissions_required() | (in module <i>fobi.contrib.plugins.form_elements.fields.file.forms.FileInputForm</i> attribute), 256 |
| phantom_js_clean_up() | (in module <i>fobi.tests.helpers</i>), 364 |
| plugin_data | (<i>fobi.models.BaseAbstractPluginEntry</i> attribute), 420 |
| plugin_data_fields | (<i>fobi.base.BaseFormFieldPluginForm</i> attribute), 378 |
| plugin_data_fields | (<i>fobi.base.BasePluginForm</i> attribute), 383 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.content.content_image</i> attribute), 222 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.content.content_image</i> attribute), 225 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.content.content_markdown</i> attribute), 228 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.content.content_richtext</i> attribute), 231 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.content.content_text</i> attribute), 234 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.content.content_video</i> attribute), 236 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.boolean</i> attribute), 238 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.checkbox_select</i> attribute), 241 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.date</i> attribute), 243 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.date_drop_down</i> attribute), 245 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.datetime</i> attribute), 247 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.decimal</i> attribute), 249 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.duration</i> attribute), 251 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.email</i> attribute), 253 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.file</i> attribute), 256 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.float</i> attribute), 258 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.hidden</i> attribute), 260 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.input</i> attribute), 261 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.integer</i> attribute), 263 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.ip_address</i> attribute), 265 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.null_boolean</i> attribute), 266 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.password</i> attribute), 268 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.range_select</i> attribute), 273 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.regex</i> attribute), 275 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_model_object</i> attribute), 280 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_mptt_model_obj</i> attribute), 282 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_multiple</i> attribute), 285 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_multiple_model</i> attribute), 287 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_m</i> attribute), 290 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_multiple_with_m</i> attribute), 293 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_multiple_with_m</i> attribute), 293 |
| plugin_data_fields | (<i>fobi.contrib.plugins.form_elements.fields.select_multiple_with_m</i> attribute), 293 |

| Module | Class | Method | Attribute | Value |
|--|--|--|-----------|---|
| (fobi.contrib.plugins.form_elements.fields.slider.forms.SliderForm) | plugin_data_repr() | | | 329 |
| plugin_data_fields | (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailForm) | | | 329 |
| (fobi.contrib.plugins.form_elements.fields.slug.forms.SlugInputForm) | plugin_name() | (fobi.models.BaseAbstractPluginEntry | | method), 420 |
| plugin_data_fields | (fobi.contrib.plugins.form_elements.fields.text.forms.TextInputForm) | plugin_not_found_error_message | | (fobi.base.BaseRegistry attribute), 384 |
| (fobi.contrib.plugins.form_elements.fields.textarea.forms.TextAreaForm) | plugin_not_found_exception_cls | (fobi.base.BaseRegistry attribute), 384 | | |
| plugin_data_fields | plugin_not_found_exception_cls | (fobi.base.FormElementPluginRegistry | at- | tribute), 387 |
| (fobi.contrib.plugins.form_elements.fields.time.forms.TimeInputForm) | plugin_not_found_exception_cls | (fobi.base.FormHandlerPluginRegistry | at- | tribute), 388 |
| plugin_data_fields | plugin_not_found_exception_cls | (fobi.base.FormWizardHandlerPluginRegistry | | tribute), 388 |
| (fobi.contrib.plugins.form_elements.security.captcha.forms.CaptchaForm) | plugin_not_found_exception_cls | | | |
| plugin_data_fields | (fobi.base.IntegrationFormElementPluginRegistry | | | |
| (fobi.contrib.plugins.form_elements.security.honeypot.forms.HoneypotForm) | plugin_not_found_exception_cls | | | |
| plugin_data_fields | (fobi.base.IntegrationFormHandlerPluginRegistry | | | |
| (fobi.contrib.plugins.form_elements.security.recaptcha.forms.ReCaptchaForm) | plugin_uid() | (fobi.contrib.plugins.form_elements.content.content_markdown | | tribute), 228 |
| plugin_data_fields | (fobi.contrib.plugins.form_handlers.http_repost.forms.HTTPRepostForm) | (fobi.contrib.plugins.form_elements.content.content_richtext | | tribute), 231 |
| plugin_data_fields | plugin_uid() | (fobi.contrib.plugins.form_elements.fields.date.widgets.Base | | tribute), 243 |
| (fobi.contrib.plugins.form_handlers.mail.forms.MailForm) | plugin_uid() | (fobi.contrib.plugins.form_elements.fields.datetime.widgets. | | tribute), 247 |
| plugin_data_repr() | (fobi.base.BasePlugin | | | tribute), 296 |
| plugin_data_repr() | (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostHandlerPlugin | | | tribute), 312 |
| plugin_data_repr() | plugin_uid() | (fobi.contrib.plugins.form_handlers.db_store.widgets.BaseD | | tribute), 416 |
| (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostWizardHandlerPlugin | plugin_uid() | (fobi.models.FormElement attribute), 416 | | |
| plugin_data_repr() | plugin_uid() | (fobi.models.FormElementEntry at- | | |
| (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostHandlerPlugin | plugin_uid() | (fobi.models.FormHandler attribute), 417 | | |
| plugin_data_repr() | plugin_uid() | (fobi.models.FormHandlerEntry at- | | |
| (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers.HTTPRepostWizardHandlerPlugin | plugin_uid() | (fobi.models.FormWizardHandler at- | | |
| plugin_data_repr() | (fobi.contrib.plugins.form_handlers.mail.base.MailHandlerPlugin | (fobi.models.FormWizardHandlerEntry | | tribute), 429 |
| plugin_data_repr() | plugin_uid_admin() | | | |
| (fobi.contrib.plugins.form_handlers.mail.base.MailWizardHandlerPlugin) | plugin_uid_code() | | | |
| plugin_data_repr() | (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailHandlerPluginModel | method), | | 415 |

415

`plugin_uid_code()` (*fobi.models.BaseAbstractPluginEntry method*), **420**

`PluginDoesNotExist`, **403**

`pop()` (*fobi.data_structures.SortableDict method*), **399**

`popitem()` (*fobi.data_structures.SortableDict method*), **399**

`position` (*fobi.models.FormElementEntry attribute*), **421**

`position` (*fobi.models.FormWizardFormEntry attribute*), **428**

`position_prop_name` (*fobi.contrib.plugins.form_importers.mailchimp_importer.base.MailChimpImporter attribute*), **332**

`position_prop_name` (*fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporter attribute*), **333**

`position_prop_name` (*fobi.form_importers.BaseFormImporter attribute*), **403**

`post()` (*fobi.views.FormWizardView method*), **437**

`post()` (*fobi.wizard.views.dynamic.DynamicNamedUrlWizardView method*), **254**

`post()` (*fobi.wizard.views.dynamic.DynamicWizardView method*), **370**

`post_processor()` (*fobi.base.BasePlugin method*), **381**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_image_base.ContentImagePlugin method*), **221**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_image_base.ContentImagePlugin method*), **222**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_image_base.ContentImageURLPlugin method*), **224**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_image_url_fobi_form_elements.ContentImageURLPlugin method*), **225**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_image_urls_for_content_multiple_fobi_form_elements.ContentImageURLPlugin method*), **226**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_markdown_fobi_form_elements.ContentMarkdownPlugin method*), **227**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_rich_text_base.ContentRichTextPlugin method*), **229**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_rich_text_fobi_form_elements.ContentRichTextPlugin method*), **230**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_text_base.ContentTextPlugin method*), **232**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_text_fobi_form_elements.ContentTextPlugin method*), **233**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_video_base.ContentVideoPlugin method*), **234**

`post_processor()` (*fobi.contrib.plugins.form_elements.content_content_video_fobi_form_elements.ContentVideoPlugin method*), **235**

`post_processor()` (*fobi.contrib.plugins.form_elements.test.dummy_plugin_integration.base.DRFIntegrationFormElement method*), **312**

`post_processor()` (*fobi.contrib.plugins.form_elements.test.dummy_plugin_integration.base.DRFIntegrationFormElement method*), **312**

`pre_processor()` (*fobi.base.BasePlugin method*), **382**

`prep_value()` (*fobi.contrib.plugins.form_handlers.mail.widgets.MultiEntryFieldWidget method*), **331**

`prepare_form_entry_export_data()` (*in module fobi.utils*), **432**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_fobi_form_elements.method*), **239**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_fobi_form_elements.method*), **239**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.duration.base.DurationField method*), **250**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.file.base.FileInputPlugin method*), **255**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.file_fobi_form_elements.method*), **255**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.select_model_object_base.SelectModelObjectPlugin method*), **278**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.select_model_object_fobi_form_elements.SelectModelObjectPlugin method*), **279**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.select_multiple_base.SelectMultiplePlugin method*), **282**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.select_multiple_fobi_form_elements.SelectMultiplePlugin method*), **284**

`prepare_plugin_form_data()` (*fobi.contrib.plugins.form_elements.fields.select_multiple_with_n_fobi_form_elements.SelectMultipleWithNPlugin method*), **292**

`process()` (*fobi.base.BasePlugin method*), **382**

`process()` (*fobi.contrib.plugins.form_elements.goldenrich_text_plugin.integration.base.DRFIntegrationFormElement method*), **210**

process_plugin_data() (fobi.base.BasePlugin method), 382

process_step() (fobi.wizard.views.dynamic.DynamicWizardView method), 370

process_step_files() (fobi.wizard.views.dynamic.DynamicWizardView method), 370

Q

quantize() (fobi.contrib.plugins.form_elements.fields.decimal.DecimalFieldInputForm method), 249

queryset (fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet attribute), 215

R

radio_fields (fobi.admin.FormEntryAdmin attribute), 375

radio_fields (fobi.admin.FormWizardEntryAdmin attribute), 377

RadioInputForm (class in fobi.contrib.plugins.form_elements.fields.radio.forms), 270

RadioInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.radio.base), 190

RadioInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.regex.base), 190

RadioInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_form_elements), 192

RadioInputPlugin (class in fobi.contrib.plugins.form_elements.fields.radio.base), 268

RadioInputPlugin (class in fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements), 269

RangeSelectInputForm (class in fobi.contrib.plugins.form_elements.fields.range_select.forms), 273

RangeSelectInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.range_select.base), 191

RangeSelectInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.range_select.fobi_form_elements), 191

RangeSelectInputPlugin (class in fobi.contrib.plugins.form_elements.fields.range_select.base), 271

RangeSelectInputPlugin (class in fobi.contrib.plugins.form_elements.fields.range_select.fobi_form_elements), 272

read_only_fields (fobi.contrib.apps.drf_integration.serializers.FobiFormEntrySerializer attribute), 214

readonly_fields (fobi.admin.BasePluginModelAdmin attribute), 374

readonly_fields (fobi.admin.FormElementEntryAdmin attribute), 374

readonly_fields (fobi.admin.FormEntryAdmin attribute), 375

readonly_fields (fobi.admin.FormHandlerEntryAdmin attribute), 376

readonly_fields (fobi.admin.FormWizardEntryAdmin attribute), 377

readonly_fields (fobi.contrib.plugins.form_handlers.db_store.admin attribute), 314

readonly_fields (fobi.contrib.plugins.form_handlers.db_store.admin attribute), 315

ReCaptchaInputForm (class in fobi.contrib.plugins.form_elements.security.recaptcha.forms), 311

ReCaptchaInputPlugin (class in fobi.contrib.plugins.form_elements.security.recaptcha.base), 310

ReCaptchaInputPlugin (class in fobi.contrib.plugins.form_elements.security.recaptcha.fobi_form_elements), 310

RegexInputForm (class in fobi.contrib.plugins.form_elements.fields.regex.forms), 275

RegexInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.regex.base), 192

RegexInputPlugin (class in fobi.contrib.apps.drf_integration.form_elements.fields.regex.fobi_form_elements), 192

RegexInputPlugin (class in fobi.contrib.plugins.form_elements.fields.regex.base), 274

RegexInputPlugin (class in fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements), 274

register() (fobi.base.BaseRegistry method), 384

register() (fobi.base.FormCallbackRegistry method), 385

register() (fobi.base.IntegrationFormCallbackRegistry method), 394

registry (fobi.base.BaseRegistry attribute), 384

registry (fobi.base.IntegrationFormCallbackRegistry attribute), 394

render() (fobi.base.BasePlugin method), 382

render() (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget method), 218

render() (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget method), 331

render() (fobi.reusable.markdown_widget.widgets.MarkdownWidget method), 360

render() (fobi.widgets.RichSelect method), 438

render() (fobi.wizard.views.dynamic.DynamicWizardView method), 370

| | | | |
|--|---|--|---|
| render_auth_link() | (in module run()) | (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.D | |
| fobi.templatetags.fobi_tags), 361 | | method), 317 | |
| render_done() | (fobi.views.FormWizardView run()) | (fobi.contrib.plugins.form_handlers.db_store.fobi_form_handlers.D | |
| method), 437 | | method), 318 | |
| render_done() | (fobi.wizard.views.dynamic.DynamicNamedUrlWizardView run()) | (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostH | |
| method), 372 | | method), 323 | |
| render_done() | (fobi.wizard.views.dynamic.DynamicWizardView run()) | (fobi.contrib.plugins.form_handlers.http_repost.base.HTTPRepostV | |
| method), 371 | | method), 323 | |
| render_fobi_forms_list() | (in module run()) | (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers | |
| fobi.templatetags.fobi_tags), 361 | | method), 324 | |
| render_goto_step() | run() | (fobi.contrib.plugins.form_handlers.http_repost.fobi_form_handlers | |
| (fobi.wizard.views.dynamic.DynamicNamedUrlWizardView method), 372 | | method), 325 | |
| method), 372 | | run() | (fobi.contrib.plugins.form_handlers.mail.base.MailHandlerPlugin |
| render_goto_step() | | method), 326 | |
| (fobi.wizard.views.dynamic.DynamicWizardView run()) | | (fobi.contrib.plugins.form_handlers.mail.base.MailWizardHandlerF | |
| method), 371 | | method), 327 | |
| render_next_step() | run() | (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailF | |
| (fobi.wizard.views.dynamic.DynamicNamedUrlWizardView method), 372 | | method), 329 | |
| method), 372 | | run() | (fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers.MailV |
| render_next_step() | | method), 329 | |
| (fobi.wizard.views.dynamic.DynamicWizardView run_form_handlers() (in module fobi.base), 396 | | method), 371 | |
| method), 371 | | run_form_handlers() (in module | |
| render_revalidation_failure() | | fobi.contrib.apps.drf_integration.base), 211 | |
| (fobi.wizard.views.dynamic.DynamicNamedUrlWizardView form_wizard_handlers() (in module | | method), 372 | |
| method), 372 | | fobi.base), 396 | |
| render_revalidation_failure() | run_integration_handler() | | |
| (fobi.wizard.views.dynamic.DynamicWizardView (fobi.base.ModelFormPlugin method), | | 388 | |
| method), 371 | | | |
| required (fobi.base.BaseFormFieldPluginForm run_validation() (fobi.contrib.apps.drf_integration.fields.NoneField | | attribute), 378 | |
| method), 213 | | | |
| reset_sequences (fobi.tests.base.BaseFobiBrowserBuldDynamicFormsTest | | | |
| attribute), 363 | | | |
| RichSelect (class in fobi.widgets), 438 | | | |
| RichSelectInverseQuotes (class in fobi.widgets), 438 | | safe_text() (in module fobi.helpers), 412 | |
| | | save_form_data_entry() | |
| run () (fobi.base.ModelFormPlugin method), 388 | | (fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandl | |
| run () (fobi.base.ModelFormWizardHandlerPlugin method), 389 | | method), 315 | |
| run () (fobi.contrib.apps.drf_integration.form_handlers.db_store.base.DBStoreHandlerPlugin | | method), 318 | |
| method), 207 | | save_plugin_data() (fobi.base.BasePluginForm | |
| run () (fobi.contrib.apps.drf_integration.form_handlers.db_store.fobi_integration_form_handlers.DBStoreHandlerPlugin | | method), 385 | |
| method), 208 | | save_plugin_data() | |
| run () (fobi.contrib.apps.drf_integration.form_handlers.http_repost.base.HTTPRepostHandlerPlugin | | method), 222 | |
| method), 208 | | save_data (fobi.contrib.plugins.form_handlers.db_store.models.Abstra | |
| run () (fobi.contrib.apps.drf_integration.form_handlers.http_repost.fobi_integration_form_handlers.HTTPRepostHandlerPlugin | | attribute), 319 | |
| method), 208 | | SavedFormEntry (class in | |
| run () (fobi.contrib.apps.drf_integration.form_handlers.mail.base.MailHandlerPlugin | | fobi.contrib.plugins.form_handlers.db_store.models), | |
| method), 209 | | run () (fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_in | |
| method), 209 | | method), 209 | |
| run () (fobi.contrib.plugins.form_handlers.db_store.base.DBStoreHandlerPlugin | | SavedFormEntry.DoesNotExist, 319 | |
| method), 315 | | SavedFormEntry.MultipleObjectsReturned, | |
| run () (fobi.contrib.plugins.form_handlers.db_store.base.DBStoreWizardHandlerPlugin | | 319 | |
| method), 316 | | savedformentry_set (fobi.models.FormEntry | |
| | | attribute), 423 | |

SavedFormDataEntryAdmin (class in 282
 fobi.contrib.plugins.form_handlers.db_store.admin), 314
 SavedFormDataEntryAdmin.Meta (class in 280
 fobi.contrib.plugins.form_handlers.db_store.admin), 314
 SavedFormWizardDataEntry (class in 281
 fobi.contrib.plugins.form_handlers.db_store.models), 320
 SavedFormWizardDataEntry.DoesNotExist, 285
 320
 SelectMultipleInputPlugin (class in
 SavedFormWizardDataEntry.MultipleObjectsReturnfobi.contrib.apps.drf_integration.form_elements.fields.select_mulu
 320 196
 savedformwizarddataentry_set SelectMultipleInputPlugin (class in
 (fobi.models.FormWizardEntry attribute), fobi.contrib.apps.drf_integration.form_elements.fields.select_mulu
 426 196
 SavedFormWizardDataEntryAdmin (class in SelectMultipleInputPlugin (class in
 fobi.contrib.plugins.form_handlers.db_store.admin), fobi.contrib.plugins.form_elements.fields.select_multiple.base),
 314 282
 SavedFormWizardDataEntryAdmin.Meta (class SelectMultipleInputPlugin (class in
 in fobi.contrib.plugins.form_handlers.db_store.admin), fobi.contrib.plugins.form_elements.fields.select_multiple.fobi_for
 314 284
 SelectInputForm (class in SelectMultipleModelObjectsInputForm
 fobi.contrib.plugins.form_elements.fields.select.forms), (class in fobi.contrib.plugins.form_elements.fields.select_multiple
 277 287
 SelectInputPlugin (class in SelectMultipleModelObjectsInputPlugin
 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base), (class in fobi.contrib.apps.drf_integration.form_elements.fields.se
 193 197
 SelectInputPlugin (class in SelectMultipleModelObjectsInputPlugin
 fobi.contrib.apps.drf_integration.form_elements.fields.select_multiple.base), (class in fobi.contrib.apps.drf_integration.form_elements.fields.se
 193 198
 SelectInputPlugin (class in SelectMultipleModelObjectsInputPlugin
 fobi.contrib.plugins.form_elements.fields.select.base), (class in fobi.contrib.plugins.form_elements.fields.select_multiple
 275 286
 SelectInputPlugin (class in SelectMultipleModelObjectsInputPlugin
 fobi.contrib.plugins.form_elements.fields.select.fobi_form_element), (class in fobi.contrib.plugins.form_elements.fields.select_multiple
 276 287
 SelectModelObjectInputForm (class in SelectMultipleMPTTModelObjectsInputForm
 fobi.contrib.plugins.form_elements.fields.select_model_object_base), (class in fobi.contrib.plugins.form_elements.fields.select_multiple
 280 290
 SelectModelObjectInputPlugin (class in SelectMultipleMPTTModelObjectsInputPlugin
 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object_base), (class in fobi.contrib.plugins.form_elements.fields.select_multiple
 194 288
 SelectModelObjectInputPlugin (class in SelectMultipleMPTTModelObjectsInputPlugin
 fobi.contrib.apps.drf_integration.form_elements.fields.select_model_object_base), (class in fobi.contrib.plugins.form_elements.fields.select_multiple
 195 289
 SelectModelObjectInputPlugin (class in SelectMultipleWithMaxInputForm (class in
 fobi.contrib.plugins.form_elements.fields.select_model_object_base), (class in fobi.contrib.plugins.form_elements.fields.select_multiple_with_m
 278 293
 SelectModelObjectInputPlugin (class in SelectMultipleWithMaxInputPlugin (class in
 fobi.contrib.plugins.form_elements.fields.select_model_object_base), (class in fobi.contrib.plugins.form_elements.fields.select_mulu
 279 199
 SelectMPTTModelObjectInputForm (class in SelectMultipleWithMaxInputPlugin (class in
 fobi.contrib.plugins.form_elements.fields.select_mptt_model_object_base), (class in fobi.contrib.apps.drf_integration.form_elements.fields.select_mulu

200
 SelectMultipleWithMaxInputPlugin (class in *fobi.contrib.plugins.form_elements.fields.slider:fobi_form_element*
fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.base), 295
 290
 SelectMultipleWithMaxInputPlugin (class in *fobi.contrib.themes.bootstrap3.widgets.form_elements.slider_bootstrap3*
fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements), 299
 292
 slug (*fobi.models.FormEntry* attribute), 423
 send_email () (*fobi.contrib.plugins.form_handlers.mail.callbacks.FobiAutoFormMailWizardEntry* attribute), 427
 method), 328
 send_email () (*fobi.contrib.plugins.form_handlers.mail.mixins.MixinMailMixin* attribute), 427
 method), 331
 send_mail () (in module *SlugInputPlugin* (class in *fobi.contrib.apps.drf_integration.form_elements.fields.slug.base*),
 330 202
 SessionWizardView (class in *SlugInputPlugin* (class in *fobi.contrib.apps.drf_integration.form_elements.fields.slug:fobi_i*
fobi.wizard.views.views), 373
 setdefault () (*fobi.data_structures.SortableDict* (class in *fobi.data_structures*), 398
 method), 399
 setUp () (*fobi.tests.test_core.FobiCoreTest* method), 365
 setUp () (*fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTests* (class in *fobi.contrib.plugins.form_elements.fields.slug:fobi_form_element*
 method), 365
 setUp () (*fobi.tests.test_dynamic_forms.FobiDynamicFormsTest* (class in *fobi.contrib.plugins.form_elements.fields.slug:fobi_form_element*
 method), 366
 setUp () (*fobi.tests.test_feincms_integration.FeinCMSIntegrationTests* (class in *fobi.contrib.plugins.form_handlers.db_store.callbacks.AutoFormD*
 method), 366
 setUp () (*fobi.tests.test_form_importers_mailchimp.FormImporterMailchimpFormImporter* (class in *fobi.contrib.plugins.form_handlers.mail.callbacks.AutoFormMail*
 method), 366
 setUp () (*fobi.tests.test_sortable_dict.FobiDataStructuresTest* (class in *fobi.data_structures*), 398
 method), 367
 setUp_app () (in module *fobi.tests.helpers*), 364
 setUpClass () (*fobi.tests.base.BaseFobiBrowserBulDynamicFormTests* (class in *fobi.tests*), 387
 class method), 363
 setUpTestData () (*fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTests* (class in *fobi.contrib.plugins.form_elements.fields.slug:fobi_form_element*
 class method), 365
 show_all_navigation_buttons
 (*fobi.models.FormWizardEntry* attribute), 427
 SimpleTheme (class in *fobi.contrib.themes.simple.fobi_themes*), 353
 skip () (in module *fobi.tests.core*), 363
 SliderInputForm (class in *fobi.contrib.plugins.form_elements.fields.slider:form*), 295
 SliderInputPlugin (class in *storage_name* (*fobi.wizard.views.dynamic.DynamicCookieWizardView*
fobi.contrib.apps.drf_integration.form_elements.fields.slider:base), 201
 SliderInputPlugin (class in *storage_name* (*fobi.wizard.views.dynamic.DynamicNamedUrlCookieWi*
fobi.contrib.apps.drf_integration.form_elements.fields.slider:fobi_integration.drf_integration), 201
 SliderInputPlugin (class in *storage_name* (*fobi.wizard.views.dynamic.DynamicNamedUrlSessionWi*
fobi.contrib.plugins.form_elements.fields.slider:fobi_integration.drf_integration), 201
 SliderInputPlugin (class in *storage_name* (*fobi.wizard.views.dynamic.DynamicSessionWizardView*
fobi.contrib.plugins.form_elements.fields.slider:base), 294
 StrippedRequest (class in *fobi.helpers*), 412

StrippedUser (class in fobi.helpers), 413

submit_plugin_form_data() (fobi.base.FormElementPlugin method), 386

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.base.DRFSubmitPluginFormElementMixin method), 210

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_multiple_plugin.base.CheckboxSelectMultiplePlugin method), 173

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.checkbox_select_plugin.fobi_integration.form_elements.fields.checkbox_select_plugin method), 174

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.duration.base.DurationPlugin method), 179

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.duration.base.DurationPlugin method), 179

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.file.base.FileInputPlugin method), 181

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.file.base.FileInputPlugin method), 182

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 194

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 195

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 196

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 196

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 197

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 198

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 199

submit_plugin_form_data() (fobi.contrib.apps.drf_integration.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 200

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_plugin.base.CheckboxSelectMultiplePlugin method), 239

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple_plugin.base.CheckboxSelectMultiplePlugin method), 240

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.date.base.DateInputPlugin method), 242

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.date.fobi_form_element method), 242

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.datetime.base.DateTimeField method), 246

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.datetime.base.DateTimeField method), 246

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.duration.base.DurationPlugin method), 250

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.duration.base.DurationPlugin method), 250

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 254

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 255

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 269

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 270

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 276

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 277

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 278

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 279

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_model_obj_field_plugin.FileInputPlugin method), 281

method), 281

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.base.SelectMultipleModelObjectsInputPlugin method), 283

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements.SelectMultipleModelObjectsInputPlugin method), 284

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements.SelectMultipleModelObjectsInputPlugin method), 286

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements.SelectMultipleModelObjectsInputPlugin method), 287

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements.SelectMultipleModelObjectsInputPlugin method), 288

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.fobi_form_elements.SelectMultipleModelObjectsInputPlugin method), 289

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements.SelectMultipleWithMaxInputPlugin method), 291

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.select_multiple_with_max.fobi_form_elements.SelectMultipleWithMaxInputPlugin method), 292

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin method), 302

submit_plugin_form_data() (fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin method), 302

submit_plugin_form_data() (in module fobi.base), 396

submit_plugin_form_data() (in module fobi.contrib.apps.drf_integration.base), 211

success_page_message (fobi.models.FormEntry attribute), 423

success_page_message (fobi.models.FormWizardEntry attribute), 427

success_page_template_name (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 218

success_page_text (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 218

success_page_title (fobi.contrib.apps.feincms_integration.widgets.FobiFormWidget attribute), 218

success_page_title (fobi.models.FormEntry attribute), 423

success_page_title (fobi.models.FormWizardEntry attribute), 427

suffix (fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet attribute), 215

multiple_base.SelectMultipleInputPlugin), 432

T

tearDown() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegration method), 363

tearDownClass() (fobi.tests.base.BaseFobiBrowserBuldDynamicForm class method), 363

template_name (fobi.wizard.views.dynamic.DynamicWizardView attribute), 438

template_name (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegration method), 363

templates (fobi.contrib.plugins.form_importers.mailchimp_importer.base attribute), 332

templates (fobi.contrib.plugins.form_importers.mailchimp_importer.fobi_form_elements.SelectMultipleMailchimpImporter method), 333

templates (fobi.form_importers.BaseFormImporter attribute), 433

test_01_assemble_form_class_and_render_form() (fobi.tests.test_dynamic_forms.FobiDynamicFormsTest method), 366

test_01_get_registered_form_element_plugins() (fobi.tests.test_core.FobiCoreTest method), 365

test_01_put_action_public_form() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegration method), 365

test_01_sortable_dict_move_before_key() (fobi.tests.test_sortable_dict.FobiDataStructuresTest method), 367

test_01_test_mailchimp_importer() (fobi.tests.test_form_importers_mailchimp.FormImportersMailchimp method), 367

test_02_get_registered_form_handler_plugins() (fobi.tests.test_core.FobiCoreTest method), 365

test_02_put_action_public_form() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegration method), 365

test_02_sortable_dict_move_after_key() (fobi.tests.test_sortable_dict.FobiDataStructuresTest method), 367

test_03_fail_put_action_public_form() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegration method), 366

test_03_get_registered_form_callbacks() (fobi.tests.test_core.FobiCoreTest method), 365

test_04_fail_options_action_non_public_form() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegration method), 366

test_04_get_registered_themes() (fobi.tests.test_core.FobiCoreTest method), 365

test_05_action_url() (fobi.tests.test_core.FobiCoreTest method), 364

test_05_fail_put_action() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 366

test_06_form_entry_get_absolute_url() (fobi.tests.test_core.FobiCoreTest method), 365

test_06_options_action_non_public_form_auth_use() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 366

test_07_form_wizard_entry_get_absolute_url() (fobi.tests.test_core.FobiCoreTest method), 365

test_07_put_action_non_public_form_auth_use() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 366

test_08_form_entry_is_active() (fobi.tests.test_core.FobiCoreTest method), 365

test_08_get_action_public_form() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 366

test_09_get_action_non_public_form_auth_use() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 366

test_1001_open_dashboard() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_10_fail_get_action_non_public_form() (fobi.tests.test_drf_integration.FobiDjangoRestFrameworkIntegrationTest method), 366

test_2001_add_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_2002_edit_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_2003_delete_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_2004_submit_form() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_3001_add_form_elements() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_3002_remove_form_elements() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_3003_edit_form_elements() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_4001_add_form_handlers() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_4002_remove_form_handlers() (fobi.tests.test_browser_build_dynamic_forms.FobiBrowserBuildDynamicFormsTest method), 364

test_fobi_form_widget_public_form() (fobi.tests.test_feincms_integration.FeinCMSIntegrationTest method), 365

TextareaForm (class in fobi.contrib.plugins.form_elements.fields.textarea.forms), 301

TextareaPlugin (class in fobi.contrib.plugins.drf_integration.form_elements.fields.textarea.base), 204

TextareaPlugin (class in fobi.contrib.plugins.drf_integration.form_elements.fields.textarea.fobi_form_elements), 300

TextareaPlugin (class in fobi.contrib.plugins.drf_integration.form_elements.fields.textarea.fobi_form_elements), 300

TextInputForm (class in fobi.contrib.plugins.form_elements.fields.text.forms), 299

TextInputPlugin (class in fobi.contrib.plugins.drf_integration.form_elements.fields.text.base), 203

TextInputPlugin (class in fobi.contrib.plugins.drf_integration.form_elements.fields.text.fobi_form_elements), 298

TextInputPlugin (class in fobi.contrib.plugins.drf_integration.form_elements.fields.text.fobi_form_elements), 299

theme() (in module fobi.context_processors), 397

theme_uid (fobi.contrib.themes.bootstrap3.widgets.form_elements.container attribute), 335

theme_uid (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_input attribute), 337

theme_uid (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy attribute), 337

theme_uid (fobi.contrib.themes.bootstrap3.widgets.form_elements.slider attribute), 339

theme_uid (fobi.contrib.themes.djangocms_admin_style_theme.widgets.fobi_form_elements attribute), 335

| | |
|---|--|
| attribute), 342 | |
| theme_uid (fobi.contrib.themes.djangocms_admin_style_theme_v(fobi.base.FobiHandlerPluginRegistry elements.DbStorePluginWidget attribute), 388 | |
| theme_uid (fobi.contrib.themes.foundation5.widgets.form_elements(fobi.base.FobiHandlerPluginRegistry elements.ContentForm attribute), 345 | |
| theme_uid (fobi.contrib.themes.foundation5.widgets.form_elements(fobi.base.FobiHandlerPluginRegistry elements.DatePluginWidget attribute), 346 | |
| theme_uid (fobi.contrib.themes.foundation5.widgets.form_elements(fobi.base.FobiHandlerPluginRegistry elements.DateTimePlugin attribute), 347 | |
| theme_uid (fobi.contrib.themes.foundation5.widgets.form_elements(fobi.base.FobiHandlerPluginRegistry elements.DummyPlugin attribute), 347 | |
| theme_uid (fobi.contrib.themes.foundation5.widgets.form_handlers(fobi.contrib.themes.foundation5.widgets.form_handlers(fobi.base.FobiHandlerPluginRegistry elements.DbStorePlugin attribute), 348 | |
| theme_uid (fobi.contrib.themes.simple.widgets.form_elements.content_richtext_simple_widget.fobi_form_elements.ContentRichText attribute), 351 | |
| theme_uid (fobi.contrib.themes.simple.widgets.form_handlers(fobi.base.FobiHandlerPluginRegistry elements.DbStorePluginWidget attribute), 352 | |
| ThemeDoesNotExist, 403 | |
| TimeInputForm (class in uid (fobi.contrib.apps.drif_integration.form_elements.content.content_image attribute), 166 | |
| fobi.contrib.plugins.form_elements.fields.time.forms), 303 | |
| TimeInputPlugin (class in uid (fobi.contrib.apps.drif_integration.form_elements.content.content_image attribute), 167 | |
| fobi.contrib.apps.drif_integration.form_elements.fields.time.base), 205 | |
| TimeInputPlugin (class in uid (fobi.contrib.apps.drif_integration.form_elements.content.content_image attribute), 167 | |
| fobi.contrib.apps.drif_integration.form_elements.fields.time.fobi_integration_form_elements), 205 | |
| TimeInputPlugin (class in uid (fobi.contrib.apps.drif_integration.form_elements.content.content_image attribute), 168 | |
| fobi.contrib.plugins.form_elements.fields.time.base), 302 | |
| TimeInputPlugin (class in uid (fobi.contrib.apps.drif_integration.form_elements.content.content_richtext attribute), 169 | |
| fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements), 302 | |
| title (fobi.models.FormEntry attribute), 423 | |
| title (fobi.models.FormWizardEntry attribute), 427 | |
| to_internal_value() (fobi.contrib.apps.drif_integration.fields.MultipleChoiceWithMaxField method), 213 | |
| to_internal_value() (fobi.contrib.apps.drif_integration.fields.NoneField method), 213 | |
| to_python() (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField method), 328 | |
| to_representation() (fobi.contrib.apps.drif_integration.fields.NoneField method), 213 | |
| two_dicts_to_string() (in module fobi.helpers), 413 | |
| type (fobi.base.BaseRegistry attribute), 384 | |
| type (fobi.base.FormElementPluginRegistry attribute), 387 | |
| type (fobi.base.FormElementPluginWidgetRegistry attribute), 387 | |
| type (fobi.base.FormHandlerPluginRegistry attribute), 387 | |

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibasedatetimeplugin.form_elements.fields.select_plugin.attribute\), 176](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.range_select_plugin.attribute\), 177](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.plugin.base.RegexAttribute\), 177](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.regex.fobi_integrations.attribute\), 192](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.regex.fobi_integrations.attribute\), 178](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.regex.fobi_integrations.attribute\), 178](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.select.fobi_integrations.attribute\), 179](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.select.fobi_integrations.attribute\), 193](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.date\(fobibaseDateTimeSelectPlugin.form_elements.fields.select_model_attribute\), 180](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.email\(fobibaseEmailPlugin.form_elements.fields.select_model_attribute\), 180](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.email\(fobibaseEmailPlugin.form_elements.fields.select_multiple_attribute\), 181](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.file\(fobibaseFileInputPlugin.form_elements.fields.select_multiple_attribute\), 182](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.file\(fobibaseFileInputPlugin.form_elements.fields.select_multiple_attribute\), 182](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.file\(fobibaseFileInputPlugin.form_elements.fields.select_multiple_attribute\), 183](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.file\(fobibaseFileInputPlugin.form_elements.fields.select_multiple_attribute\), 183](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.file\(fobibaseFileInputPlugin.form_elements.fields.select_multiple_attribute\), 199](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.file\(fobibaseFileInputPlugin.form_elements.fields.select_multiple_attribute\), 184](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.file\(fobibaseFileInputPlugin.form_elements.fields.slider.base.SliderAttribute\), 184](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.input\(fobibaseFormPlugin.form_elements.fields.slider.fobi_integrations.attribute\), 185](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.input\(fobibaseFormPlugin.form_elements.fields.slug.base.SlugAttribute\), 185](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.integ\(fobibaseIntegrationPlugin.form_elements.fields.slug.fobi_integrations.attribute\), 186](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.integ\(fobibaseIntegrationPlugin.form_elements.fields.text.base.TextInputAttribute\), 186](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.ip_address\(fobibaseIPAddressPlugin.form_elements.fields.text.fobi_integrations.attribute\), 187](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.ip_address\(fobibaseIPAddressPlugin.form_elements.fields.time.fobi_integrations.attribute\), 187](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.multifobibaseMultiFormPlugin.form_elements.fields.textarea.fobi_integrations.attribute\), 188](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.multifobibaseMultiFormPlugin.form_elements.fields.checkbox.fobi_integrations.attribute\), 188](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.password\(fobibasePasswordPlugin.form_elements.fields.time.fobi_integrations.attribute\), 189](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.password\(fobibasePasswordPlugin.form_elements.fields.url.fobi_integrations.attribute\), 189](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.radio\(fobibaseRadioPlugin.form_elements.fields.url.fobi_integrations.attribute\), 190](#)

[uid\(fobi.contrib.apps.drif_integration.form_elements.fields.radio\(fobibaseRadioPlugin.store.base.DBStoreAttribute\), 190](#)

`uid(fobi.contrib.apps.drf_integration.form_handlers.db_store_repository_plugins.form_elements.fields.decimal.fobi_form_elements.attribute), 208`

`uid(fobi.contrib.apps.drf_integration.form_handlers.http_repost(basic_auth_plugin_handler_plugins.fields.decimal.fobi_form_elements.attribute), 208`

`uid(fobi.contrib.apps.drf_integration.form_handlers.http_repost(public_contrib_logs_for_handlers_fields.FERRepostsFromBaseDrgationInput(attribute), 208`

`uid(fobi.contrib.apps.drf_integration.form_handlers.mail_base.MobileFrontLib.Plugins.form_elements.fields.duration.fobi_form_element(attribute), 209`

`uid(fobi.contrib.apps.drf_integration.form_handlers.mail.fobi_mail_provider_plugins.MailedHasFieldPluginil.base.EmailInputPlug(attribute), 209`

`uid(fobi.contrib.plugins.form_elements.content.content_image.fobi.Contrib.PluginRfgin_elements.fields.email.fobi_form_elements.E(attribute), 221`

`uid(fobi.contrib.plugins.form_elements.content.content_image.fobiforcontribpluginsContentImageRfields.file.base.FileInputPlug(attribute), 222`

`uid(fobi.contrib.plugins.form_elements.content.content_image(fobibase.CibplugsforRlePlugins(fields.file.fobi_form_elements.File(attribute), 224`

`uid(fobi.contrib.plugins.form_elements.content.content_image(fobifoinfoinpluginsforContentmsgeURffPluginase.FloatInputPlug(attribute), 225`

`uid(fobi.contrib.plugins.form_elements.content.content_markdown.fobibase.CibplugsMafordwaPlugins(fields.float.fobi_form_elements.Fl(attribute), 227`

`uid(fobi.contrib.plugins.form_elements.content.content_markdown.fobifoinfoinpluginsforCelementMsJkldshPluginbase.HiddenInputPlu(attribute), 227`

`uid(fobi.contrib.plugins.form_elements.content.content_richtext.fobiaContribRightTefoPhugilements.fields.hidden.fobi_form_elements.(attribute), 229`

`uid(fobi.contrib.plugins.form_elements.content.content_richtext(fobbicontribpluginsConteleRichTefutPuginput.base.InputPlug(attribute), 230`

`uid(fobi.contrib.plugins.form_elements.content.content_textbase.GintentTexrPluginform_elements.fields.input.fobi_form_elements.In(attribute), 232`

`uid(fobi.contrib.plugins.form_elements.content.content_text.fobifoinnorelaimptngGofomntetePlugins(fields.integer.base.IntegerInputPlu(attribute), 233`

`uid(fobi.contrib.plugins.form_elements.content.content_video.fobi.ContribVidegRPluginm_elements.fields.integer.fobi_form_elements.(attribute), 235`

`uid(fobi.contrib.plugins.form_elements.content.content_video(fobiforcontribpluginsContentVideoonRfields.ip_address.base.IPAddressIn(attribute), 236`

`uid(fobi.contrib.plugins.form_elements.fields.boolean.base.Bool(fobiSelectH plugins.form_elements.fields.ip_address.fobi_form_eleme(attribute), 237`

`uid(fobi.contrib.plugins.form_elements.fields.boolean.fobi_fornfelementsB pluginsSfient Plugins(fields.null_boolean.base.NullBoole(attribute), 238`

`uid(fobi.contrib.plugins.form_elements.fields.checkbox_select(fobiplechaiseCueinksforSelateMultipledUpunAlluginolean.fobi_form_elev(attribute), 239`

`uid(fobi.contrib.plugins.form_elements.fields.checkbox_select(fobiplechfobipluggingelementsCheekbfSeLqatMwbipllbasePLugiardInp(attribute), 240`

`uid(fobi.contrib.plugins.form_elements.fields.date.base.DateInpfioPluginrib.plugins.form_elements.fields.password.fobi_form_ement(attribute), 242`

`uid(fobi.contrib.plugins.form_elements.fields.date.fobi_form_elgmentsDateImpugnRPluginm_elements.fields.radio.base.RadioInputPlug(attribute), 242`

`uid(fobi.contrib.plugins.form_elements.fields.date_drop_down.fobieDoneDropDownsfopmtPlegiants.fields.radio.fobi_form_elements.R(attribute), 244`

`uid(fobi.contrib.plugins.form_elements.fields.date_drop_down.fobiforcontribpluginsPorteDropDownsfidhsPlugin_select.base.RangeSele(attribute), 245`

`uid(fobi.contrib.plugins.form_elements.fields.datetime.base.DatTimeImprailPlugins(form_elements.fields.range_select.fobi_form_elen(attribute), 246`

`uid(fobi.contrib.plugins.form_elements.fields.datetime.fobi_fornfelementsDtueTimsfInputPlugins(fields.regex.base.RegexInputPlug(attribute), 246`

uid (fobi.contrib.plugins.form_elements.fields.regex.fobi_form_elements.RegexPlugin.elements.security.captcha.base.CaptchaInput
 attribute), 274
 uid (fobi.contrib.plugins.form_elements.fields.select.base.SelectInputPlugin.elements.security.captcha.fobi_form_elements
 attribute), 276
 uid (fobi.contrib.plugins.form_elements.fields.select.fobi_form_elements.SelectPlugin.elements.security.honeypot.base.HoneypotInput
 attribute), 277
 uid (fobi.contrib.plugins.form_elements.fields.select_model_object(fobi.contrib.plugins.fields.select_model_object_plugin.elements.security.honeypot.fobi_form_elements
 attribute), 278
 uid (fobi.contrib.plugins.form_elements.fields.select_model_object(fobi.contrib.plugins.fields.select_model_object_plugin.elements.security.honeypot.fobi_form_elements
 attribute), 280
 uid (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object(fobi.contrib.plugins.fields.select_mptt_model_object_plugin.elements.security.honeypot.fobi_form_elements
 attribute), 281
 uid (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object(fobi.contrib.plugins.fields.select_mptt_model_object_plugin.elements.security.honeypot.fobi_form_elements
 attribute), 282
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 283
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 284
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 286
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 287
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 288
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 289
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 291
 uid (fobi.contrib.plugins.form_elements.fields.select_multiple_base(fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements.DummyFormPlugin.elements.test.dummy.fobi_form_elements.DummyFormPlugin
 attribute), 293
 uid (fobi.contrib.plugins.form_elements.fields.slider.base.SliderInputPlugin.elements.http_repost.fobi_form_handlers.http_repost.fobi_form_handlers.HTTPRepostWizardHandlerPlugin
 attribute), 294
 uid (fobi.contrib.plugins.form_elements.fields.slider.fobi_form_elements.SliderPlugin.elements.mail.base.MailHandlerPlugin
 attribute), 295
 uid (fobi.contrib.plugins.form_elements.fields.slug.base.SlugInputPlugin.elements.mail.base.MailWizardHandlerPlugin
 attribute), 297
 uid (fobi.contrib.plugins.form_elements.fields.slug.fobi_form_elements.SlugInputPlugin.elements.mail.fobi_form_handlers.MailWizardHandlerPlugin
 attribute), 297
 uid (fobi.contrib.plugins.form_elements.fields.text.base.TextInputPlugin.elements.mail.fobi_form_handlers.MailWizardHandlerPlugin
 attribute), 299
 uid (fobi.contrib.plugins.form_elements.fields.text.fobi_form_elements.TextInputPlugin.elements.mailchimp_importer.base.MailChimpImporterPlugin
 attribute), 299
 uid (fobi.contrib.plugins.form_elements.fields.textarea.base.TextAreaInputPlugin.elements.mailchimp_importer.fobi_form_importers.mailchimp_importer.fobi_form_importers.MailChimpImporterPlugin
 attribute), 300
 uid (fobi.contrib.plugins.form_elements.fields.textarea.fobi_form_elements.TextAreaInputPlugin.elements.bootstrap3.fobi_themes.Bootstrap3Theme
 attribute), 301
 uid (fobi.contrib.plugins.form_elements.fields.time.base.TimeInputPlugin.elements.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme
 attribute), 302
 uid (fobi.contrib.plugins.form_elements.fields.time.fobi_form_elements.TimeInputPlugin.elements.foundation5.fobi_themes.Foundation5Theme
 attribute), 303
 uid (fobi.contrib.plugins.form_elements.fields.url.base.URLInputPlugin.elements.simple.fobi_themes.SimpleTheme
 attribute), 304
 uid (fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements.URLInputPlugin.elements.fobi_form_importers.fobi_form_importers.FobiFormImporterBaseFormImporter
 attribute), 304

[uidfy\(\)](#) (*fobi.base.FormCallbackRegistry* method), 385
[uidfy\(\)](#) (*fobi.base.IntegrationFormCallbackRegistry* method), 394
[uniquify_sequence\(\)](#) (in module *fobi.helpers*), 414
[unregister\(\)](#) (*fobi.base.BaseRegistry* method), 384
[update\(\)](#) (*fobi.contrib.apps.drf_integration.views.FobiFormEntryViewSet* method), 215
[update\(\)](#) (*fobi.data_structures.SortableDict* method), 399
[update_plugin_data\(\)](#) (*fobi.base.BasePlugin* method), 382
[update_plugin_data\(\)](#) (in module *fobi.helpers*), 414
[updated](#) (*fobi.models.FormEntry* attribute), 423
[updated](#) (*fobi.models.FormWizardEntry* attribute), 427
[url_exists\(\)](#) (in module *fobi.validators*), 432
[url_name](#) (*fobi.wizard.views.dynamic.DynamicNamedUrlWizardView* attribute), 372
[URLInputForm](#) (class in *fobi.contrib.plugins.form_elements.fields.url.forms*), 305
[URLInputPlugin](#) (class in *fobi.contrib.apps.drf_integration.form_elements.fields.url.base*), 206
[URLInputPlugin](#) (class in *fobi.contrib.apps.drf_integration.form_elements.fields.url.fobi_integration.form_elements*), 206
[URLInputPlugin](#) (class in *fobi.contrib.plugins.form_elements.fields.url.base*), 304
[URLInputPlugin](#) (class in *fobi.contrib.plugins.form_elements.fields.url.fobi_form_elements*), 304
[user](#) (*fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataEntry* attribute), 319
[user](#) (*fobi.contrib.plugins.form_handlers.db_store.models.SavedFormDataEntry* attribute), 320
[user](#) (*fobi.contrib.plugins.form_handlers.db_store.models.SavedFormWizardDataEntry* attribute), 321
[user](#) (*fobi.models.FormEntry* attribute), 424
[user](#) (*fobi.models.FormWizardEntry* attribute), 427
[user_id](#) (*fobi.contrib.plugins.form_handlers.db_store.models.AbstractSavedFormDataEntry* attribute), 319
[user_id](#) (*fobi.models.FormEntry* attribute), 424
[user_id](#) (*fobi.models.FormWizardEntry* attribute), 427
[users](#) (*fobi.models.AbstractPluginModel* attribute), 415
[users](#) (*fobi.models.FormElement* attribute), 416
[users](#) (*fobi.models.FormHandler* attribute), 417
[users](#) (*fobi.models.FormWizardHandler* attribute), 418
[users_list\(\)](#) (*fobi.models.AbstractPluginModel* method), 415
V
[validate\(\)](#) (*fobi.contrib.plugins.form_elements.fields.select_multiple_w* method), 292
[validate\(\)](#) (*fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailF* method), 329
[validate_form_element_plugin_uid\(\)](#) (in module *fobi.base*), 396
[validate_form_handler_plugin_uid\(\)](#) (in module *fobi.base*), 396
[validate_form_wizard_handler_plugin_uid\(\)](#) (in module *fobi.base*), 397
[validate_initial_for_choices\(\)](#) (in module *fobi.helpers*), 414
[validate_initial_for_multiple_choices\(\)](#) (in module *fobi.helpers*), 414
[validate_integration_form_element_plugin_uid\(\)](#) (in module *fobi.base*), 397
[validate_integration_form_handler_plugin_uid\(\)](#) (in module *fobi.base*), 397
[validate_plugin_data\(\)](#) (*fobi.base.BaseFormFieldPluginForm* method), 378
[validate_plugin_data\(\)](#) (*fobi.base.BasePluginForm* method), 383
[validate_submit_value_as\(\)](#) (in module *fobi.helpers*), 414
[validate_theme_uid\(\)](#) (in module *fobi.base*), 397
[value_for_index\(\)](#) (*fobi.data_structures.SortableDict* method), 399
[values\(\)](#) (*fobi.data_structures.SortableDict* method), 399
[view_embed_form_entry_ajax_template](#) (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 341
[view_entries_icon_class](#) (*fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStore* attribute), 322
[view_entries_icon_class](#) (*fobi.contrib.themes.djangocms_admin_style_theme.widgets.form* attribute), 342
[view_entries_icon_class](#) (*fobi.contrib.themes.foundation5.widgets.form_handlers.db_store* attribute), 348
[view_entries_icon_class](#) (*fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi* attribute), 352
[view_form_entry\(\)](#) (in module *fobi.views*), 437
[view_form_entry_ajax_template](#) (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 341
[view_form_entry_ajax_template](#) (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes* attribute), 344

[view_form_entry_ajax_template](#) (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 329
[view_form_entry_ajax_template](#) (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 350
[view_form_entry_ajax_template](#) (*fobi.contrib.plugins.form_importers.mailchimp_importer.base.MailchimpImporter* attribute), 332
[view_form_entry_ajax_template](#) (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 333
[view_form_entry_ajax_template](#) (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 354
[view_form_entry_ajax_template](#) (*fobi.form_importers.BaseFormImporter* attribute), 403
[view_form_entry_template](#) (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 341
[view_form_entry_template](#) (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 427
[view_form_entry_template](#) (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme* attribute), 372
[view_form_entry_template](#) (*fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes.DjangoCMSAdminStyleTheme* attribute), 344
[view_form_entry_template](#) (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 350
[view_form_entry_template](#) (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 354
[view_form_wizard_entry_ajax_template](#) (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 341
[view_form_wizard_entry_ajax_template](#) (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 350
[view_form_wizard_entry_ajax_template](#) (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 354
[view_form_wizard_entry_ajax_template](#) (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 341
[view_form_wizard_entry_template](#) (*fobi.contrib.themes.bootstrap3.fobi_themes.Bootstrap3Theme* attribute), 341
[view_form_wizard_entry_template](#) (*fobi.contrib.themes.foundation5.fobi_themes.Foundation5Theme* attribute), 350
[view_form_wizard_entry_template](#) (*fobi.contrib.themes.simple.fobi_themes.SimpleTheme* attribute), 354
[view_saved_form_data_entries\(\)](#) (in module *fobi.contrib.plugins.form_handlers.db_store.views*), 321
[view_saved_form_data_entries_template_name](#) (*fobi.contrib.plugins.form_handlers.db_store.widgets.BaseDbStorePluginWidget* attribute), 322
[view_saved_form_data_entries_template_name](#) (*fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.fobi_form_elements.DbStorePluginWidget* attribute), 348
[view_saved_form_wizard_data_entries\(\)](#) (in module *fobi.contrib.plugins.form_handlers.db_store.views*), 321

W

[widget](#) (*fobi.base.BasePlugin* attribute), 382
[widget](#) (*fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotField* attribute), 308
[widget](#) (*fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField* attribute), 308