
django-fobi Documentation

Release 0.5.19

Artur Barseghyan <artur.barseghyan@gmail.com>

December 18, 2015

1	Prerequisites	3
2	Key concepts	5
3	Main features and highlights	7
4	Roadmap	9
5	Some screenshots	11
6	Demo	13
6.1	Live demo	13
6.2	Run demo locally	13
7	Quick start	15
8	Installation	17
9	Creating a new form element plugin	21
9.1	Define and register the form element plugin	21
10	Creating a new form handler plugin	25
10.1	Define and register the form handler plugin	25
11	Creating a form callback	29
12	Suggestions	31
12.1	Custom action for the form	31
12.2	When you want to customise too many things	31
13	Theming	33
13.1	Create a new theme	33
13.2	Make changes to an existing theme	36
14	Permissions	39
15	Management commands	41
16	Tuning	43

17 Bundled plugins and themes	45
17.1 Bundled form element plugins	45
17.2 Bundled form handler plugins	46
17.3 Bundled themes	47
18 HTML5 fields	49
19 Loading initial data using GET arguments	51
20 Dynamic initial values	53
21 Submitted form element plugins values	55
22 Rendering forms using third-party libraries	57
22.1 Using <i>django-crispy-forms</i>	57
22.2 Using <i>django-floppyforms</i>	57
23 Import/export forms	59
24 Available translations	61
25 Debugging	63
26 Troubleshooting	65
27 License	67
28 Support	69
29 Author	71
30 Screenshots	73
30.1 Bootstrap3 theme	73
30.2 Simple theme	88
31 Documentation	93
31.1 fobi package	93
31.2 Quick start	138
32 Indices and tables	143
Bibliography	145
Python Module Index	147

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

Prerequisites

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

Note, that Django 1.8 is not yet proclaimed to be flawlessly supported. The core and contrib packages (with no additional dependencies) have been tested against the latest stable Django 1.8 release. All tests have successfully passed, although it's yet too early to claim that Django 1.8 is fully supported.

Key concepts

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

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

Main features and highlights

- User-friendly GUI to quickly build forms.
- Large variety of *Bundled form element plugins*. Most of the Django fields are supported. *HTML5 fields* are supported as well.
- Anti-spam solutions like *CAPTCHA*, *ReCAPTCHA* or *Honeypot* come out of the box (*CAPTCHA* and *ReCAPTCHA* do require additional third-party apps to be installed).
- In addition to standard form elements, there are cosmetic (presentational) form elements (for adding a piece of text, image or a embed video) alongside standard form elements.
- Data handling in plugins (form handlers). Save the data, mail it to some address or repost it to some other endpoint. See the *Bundled form handler plugins* for more information.
- Developer-friendly API, which allows to edit existing or build new form fields and handlers without touching the core.
- Support for custom user model.
- *Theming*. There are 4 ready to use *Bundled themes*: “Bootstrap 3”, “Foundation 5”, “Simple” (with editing interface in style of Django admin) and “DjangoCMS admin style” theme (which is another simple theme with editing interface in style of *djangoCMS-admin-style*).
- Implemented *integration with FeinCMS* (in a form of a FeinCMS page widget).
- Implemented *integration with DjangoCMS* (in a form of a DjangoCMS page plugin).
- Implemented *integration with Mezzanine* (in a form of a Mezzanine page).
- Reordering of form elements using drag-n-drop.
- Data export (*db_store* form handler plugin) into XLS/CSV format.
- *Dynamic initial values* for form elements.
- Import/export forms to/from JSON format.

Roadmap

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

- 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).
- Fieldsets.

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

Some screenshots

See the documentation for some screen shots:

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

6.1 Live demo

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

Credentials:

- username: test_user
- password: test_user

6.2 Run demo locally

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

Grab the latest *django_fobi_example_app_installer.sh*:

```
$ wget https://raw.githubusercontent.com/barseghyanartur/django-fobi/stable/examples/django_fobi_example_app_installer.sh
```

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

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

Open your browser and test the app.

Dashboard:

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

Django admin interface:

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

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

Quick start

See the [quick start](#).

Installation

1. Install latest stable version from PyPI:

```
$ pip install django-fobi
```

Or latest stable version from GitHub:

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

Or latest stable version from BitBucket:

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

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

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

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

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

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

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

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

And the following to the context processors.

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

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

4. Configure URLs

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

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

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

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

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

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

Creating a new form element plugin

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

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

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

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

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

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

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

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

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

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

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

9.1 Define and register the form element plugin

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

9.1.1 path/to/sample_textarea/fobi_form_elements.py

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

Required imports.

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

Defining the Sample textarea plugin.

```
class SampleTextareaPlugin(FormFieldPlugin):
    uid = "sample_textarea"
    name = "Sample Textarea"
    form = SampleTextareaForm
    group = "Samples" # Group to which the plugin belongs to

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

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

Registering the SampleTextareaPlugin plugin.

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

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

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

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

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

- *form_entry* (`fobi.models.FormEntry`): Form entry, which is being submitted.
- *request* (`django.http.HttpRequest`): The Django HTTP request.
- *form* (`django.forms.Form`): Form object (a valid one, which contains the `cleaned_data` attribute).

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

```
def submit_plugin_form_data(self, form_entry, request, form):
    # Get the file path
    file_path = form.cleaned_data.get(self.data.name, None)
    if file_path:
        # Handle the upload
        saved_file = handle_uploaded_file(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 autodiscovered file pattern. All your form element plugins should be registered in modules named `fobi_form_elements.py`.

Required imports.

```

from django import forms
from fobi.base import BasePluginForm

```

Form for for SampleTextareaPlugin form element plugin.

```

class SampleTextareaForm(forms.Form, BasePluginForm):
    plugin_data_fields = [
        ("name", ""),
        ("label", ""),
        ("initial", ""),
        ("required", False)
    ]

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

```

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

- name

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

See the following example.

```

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

```

```
saved_image = handle_uploaded_file(FILES_UPLOAD_DIR, file_path)
self.cleaned_data['file'] = saved_image
```

9.1.3 path/to/sample_textarea/widgets.py

Required imports.

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

Defining the base plugin widget.

```
class BaseSampleTextareaPluginWidget(FormElementPluginWidget):
    # Same as ``uid`` value of the ``SampleTextareaPlugin``.
    plugin_uid = "sample_textarea"
```

9.1.4 path/to/sample_layout/fobi_form_elements.py

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

Required imports.

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

Define the theme specific plugin.

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

Register the widget.

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

9.1.5 Form element plugin final steps

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

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

Afterwards, go to terminal and type the following command.

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

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

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

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

Creating a new form handler plugin

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

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

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

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

10.1 Define and register the form handler plugin

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

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

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

10.1.1 path/to/sample_mail/fobi_form_handlers.py

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

Required imports.

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

Defining the Sample mail handler plugin.

```
class SampleMailHandlerPlugin(FormHandlerPlugin):  
    uid = "sample_mail"  
    name = _("Sample mail")  
    form = SampleMailForm
```

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

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

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

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

10.1.2 path/to/sample_mail/forms.py

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

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

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

Required imports.

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

Defining the form for Sample mail handler plugin.

```
class MailForm(forms.Form, BasePluginForm):
    plugin_data_fields = [
        ("from_name", ""),
        ("from_email", ""),
        ("to_name", ""),
        ("to_email", ""),
        ("subject", ""),
        ("body", ""),
    ]

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

```
body = forms.CharField(label=_("Body"), required = False,
                        widget=forms.widgets.Textarea)
```

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

10.1.3 Prioritise the execution order

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

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

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

10.1.4 Form handler plugin custom actions

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

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

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

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

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

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

10.1.5 Form handler plugin final steps

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

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

Afterwards, go to terminal and type the following command.

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

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

Creating a form callback

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

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

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

See the callback example below.

Required imports.

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

Define and register the callback

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

Add the callback module to INSTALLED_APPS.

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

Suggestions

12.1 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.

12.2 When you want to customise too many things

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

Theming

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

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

13.1 Create a new theme

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

```
path/to/sample_theme/
-- static
|   -- css
|       -- sample_theme.css
|   -- js
|       -- sample_theme.js
-- templates
|   -- sample_theme
|       -- _base.html
```

```
|         -- add_form_element_entry.html
|         -- ...
|         -- view_form_entry_ajax.html
-- __init__.py
-- fobi_form_elements.py
-- fobi_themes.py # Where themes are defined and registered
```

See the theme example below.

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

from fobi.base import BaseTheme, theme_registry

class SampleTheme(BaseTheme):
    """
    Sample theme.
    """
    uid = 'sample'
    name = _("Sample")

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

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

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

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

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

    form_ajax = 'sample_theme/snippets/form_ajax.html'
    form_snippet_template_name = 'sample_theme/snippets/form_snippet.html'
    form_properties_snippet_template_name = 'sample_theme/snippets/form_properties_snippet.html'
    messages_snippet_template_name = 'sample_theme/snippets/messages_snippet.html'

    add_form_element_entry_template = 'sample_theme/add_form_element_entry.html'
    add_form_element_entry_ajax_template = 'sample_theme/add_form_element_entry_ajax.html'

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

    create_form_entry_template = 'sample_theme/create_form_entry.html'
```

```

create_form_entry_ajax_template = 'bootstrap3/create_form_entry_ajax.html'

dashboard_template = 'sample_theme/dashboard.html'

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

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

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

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

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

```

Registering the SampleTheme plugin.

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

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

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

```

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

```

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

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

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

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

Below follow the properties of the theme:

- `base_edit`
- `base_view`

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

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

```
generic
-- snippets
|   -- form_ajax.html
|   -- form_edit_ajax.html
|   -- *** form_properties_snippet.html
|   -- *** form_snippet.html
|   -- --- form_edit_snippet.html (does not exist in generic templates)
|   -- --- form_view_snippet.html (does not exist in generic templates)
|   -- form_view_ajax.html
|   -- messages_snippet.html
|
-- _base.html
-- add_form_element_entry.html
-- add_form_element_entry_ajax.html
-- add_form_handler_entry.html
-- add_form_handler_entry_ajax.html
-- base.html
-- create_form_entry.html
-- create_form_entry_ajax.html
-- *** dashboard.html
-- edit_form_element_entry.html
-- edit_form_element_entry_ajax.html
-- edit_form_entry.html
-- *** edit_form_entry_ajax.html
-- edit_form_handler_entry.html
-- edit_form_handler_entry_ajax.html
-- form_entry_submitted.html
-- *** form_entry_submitted_ajax.html
-- *** theme.html
-- view_form_entry.html
-- view_form_entry_ajax.html
```

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

13.2 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.

13.2.1 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
```

13.2.2 fobi_themes.py

Overriding the “simple” theme.

```
__all__ = ('MySimpleTheme',)

from fobi.base import theme_registry

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

class MySimpleTheme(SimpleTheme):
    html_classes = ['my-simple-theme',]
    base_view_template = 'override_simple_theme/base_view.html'
    form_ajax = 'override_simple_theme/snippets/form_ajax.html'
```

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

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

13.2.3 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 %}
```

13.2.4 templates/override_simple_theme/snippets/form_ajax.html

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

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

Permissions

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

Plugin access rights management interface in Django admin

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

Management commands

There are several management commands available.

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

Tuning

There are number of Dash 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.

17.1 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.

17.1.1 Fields

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

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

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

17.1.2 Content/presentation

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

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

17.1.3 Security

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

17.1.4 MPTT fields

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

17.1.5 Test

Test plugins are made for dev purposes only.

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

17.2 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.

17.3 Bundled themes

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

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

HTML5 fields

The following HTML5 fields are supported in appropriate bundled plugins:

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

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

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

Loading initial data using GET arguments

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

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

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

Dynamic initial values

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

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

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

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

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

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

- request: Stripped HttpRequest object.
 - request.path: A string representing the full path to the requested page, not including the scheme or domain.
 - request.get_full_path(): Returns the path, plus an appended query string, if applicable.
 - request.is_secure(): Returns True if the request is secure; that is, if it was made with HTTPS.
 - request.is_ajax(): Returns True if the request was made via an XMLHttpRequest, by checking the HTTP_X_REQUESTED_WITH header for the string 'XMLHttpRequest'.
 - request.META: A stripped down standard Python dictionary containing the available HTTP headers.
 - * HTTP_ACCEPT_ENCODING: Acceptable encodings for the response.
 - * HTTP_ACCEPT_LANGUAGE: Acceptable languages for the response.
 - * HTTP_HOST: The HTTP Host header sent by the client.

- * HTTP_REFERER: The referring page, if any.
- * HTTP_USER_AGENT: The client's user-agent string.
- * QUERY_STRING: The query string, as a single (unparsed) string.
- * REMOTE_ADDR: The IP address of the client.
- request.user: Authenticated user.
 - * request.user.email:
 - * request.user.get_username(): Returns the username for the user. Since the User model can be swapped out, you should use this method instead of referencing the username attribute directly.
 - * request.user.get_full_name(): Returns the first_name plus the last_name, with a space in between.
 - * request.user.get_short_name(): Returns the first_name.
 - * request.user.is_anonymous():
- now: datetime.datetime.now()
- today: datetime.date.today()

Submitted form element plugins values

While some values of form element plugins are submitted as is, some others need additional processing. There are 3 behaviours 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).

22.1 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 %}
```

22.2 Using *django-floppyforms*

```
{% load floppyforms fobi_tags %}

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

{% form form %}
```

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

Import/export forms

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

Available translations

English is the primary language.

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

Debugging

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

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

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

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

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

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

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

Troubleshooting

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

License

GPL 2.0/LGPL 2.1

Support

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

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

Screenshots

30.1 Bootstrap3 theme

30.1.1 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](#)

30.1.2 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

30.1.3 View/edit form

Form elements

Build your forms
Edit
View

Form My test form was created successfully.

Edit form

Elements
Handlers
Properties

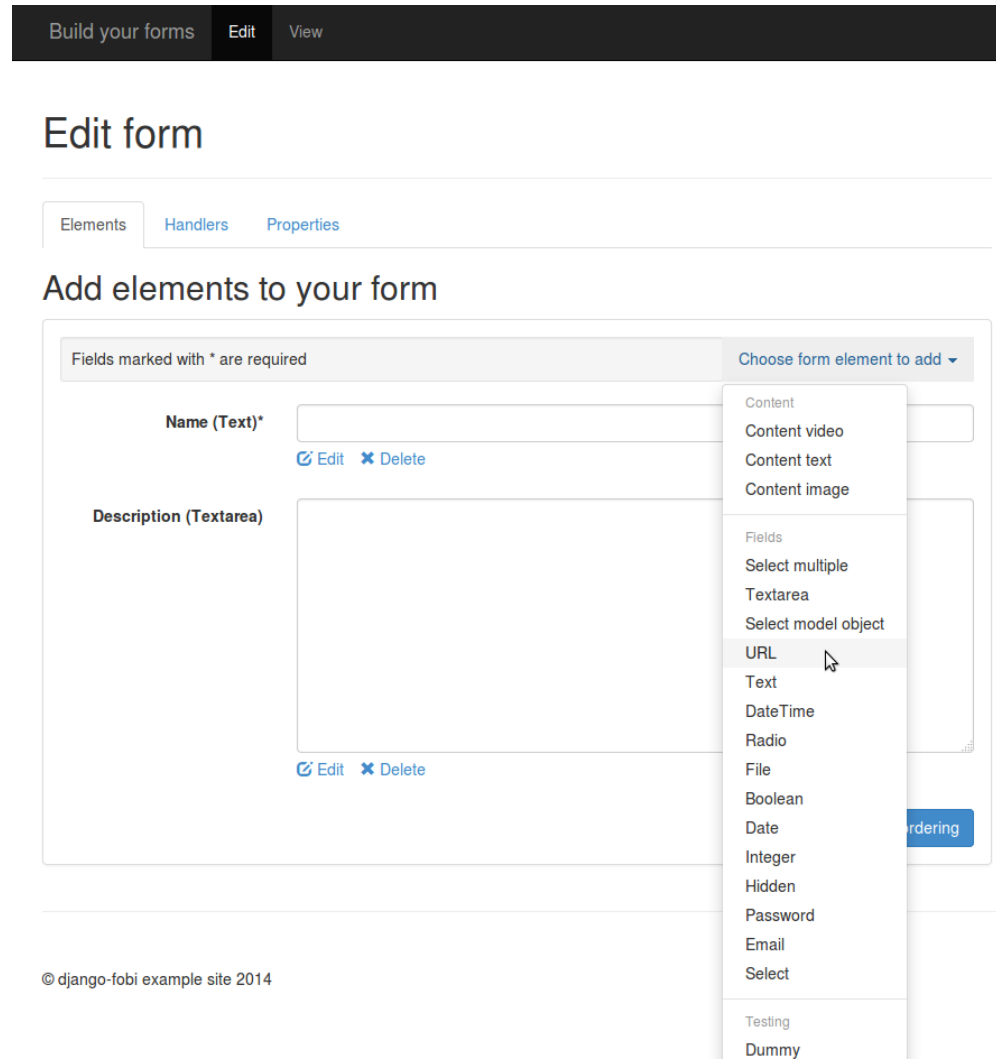
Add elements to your form

Fields marked with * are required

Choose form element to add ▾

Save ordering

© django-fobi example site 2014



Build your forms

Add "URL" element to the form

Fields marked with * are required

Label*

Name*

Help text

Initial

Max length*

Required

☐

Add

© django-fobi example site 2014

Build your forms Edit View

Edit form

Elements Handlers Properties

Add elements to your form

Fields marked with * are required [Choose form element to add ▾](#)

Name (Text)*

[✎ Edit](#) [✕ Delete](#)

Description (Textarea)

[✎ Edit](#) [✕ Delete](#)

Website (URL)

[✎ Edit](#) [✕ Delete](#)

Save ordering

© django-fobi example site 2014

Form handlers



Edit form



Add handlers to your form



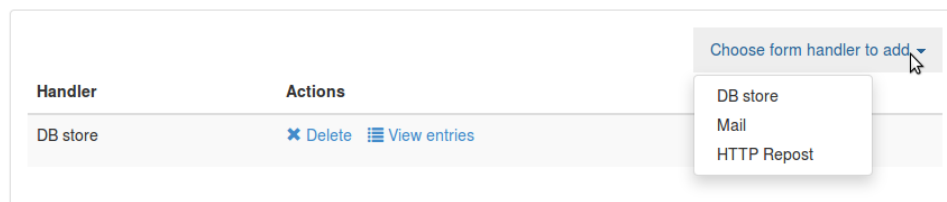
© django-fobi example site 2014



Edit form



Add handlers to your form



© django-fobi example site 2014

Build your forms

Add "Mail" handler to the form

Fields marked with * are required

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

Add

© django-fobi example site 2014

Build your forms Edit View

The form handler plugin "Mail" was added successfully.

Edit form

Elements Handlers Properties

Add handlers to your form

Choose form handler to add ▾

Handler	Actions
DB store	✕ Delete ≡ View entries
Mail ?	🔗 Edit ✕ Delete

© django-fobi example site 2014

Build your forms Edit View

Edit form

Elements Handlers Properties

Form properties

Fields marked with * are required

Name*

My test form

☒ Public? ?

Success page title

Thank you!

Custom message title to display after valid form is submitted

Success page body

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

Custom message text to display after valid form is submitted

Submit changes

© django-fobi example site 2014

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

View form

Fields marked with * are required

Name*

n.n.

Description

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

Website

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

Submit

© django-fobi example site 2014

[Build your forms](#)

Form My test form was submitted successfully.

Thank you!

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

© django-fobi example site 2014

Build your forms

Add "Content video" element to the form

Fields marked with * are required

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

© django-fobi example site 2014

Build your forms

Add "Boolean" element to the form

Fields marked with * are required

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

© django-fobi example site 2014

Build your formsEditView

Elements ordering edited successfully.

Edit form

ElementsHandlersProperties

Add elements to your form

Fields marked with * are requiredChoose form element to add ▾

(Content video)

Delusional Insanity - To Far Beyond



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

Name (Text)*

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

Description (Textarea)

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

Website (URL)

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

(Content image)



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

Image (File)

Browse... No file selected.

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

Agree? (Boolean)

☐

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

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

30.1. Bootstrap3 theme

Save ordering

85

© django-fobi example site 2014

Build your forms Edit View

View form

Fields marked with * are required



Name*

Description

Website



Image

No file selected.

Agree?

☐

30.2 Simple theme

30.2.1 View/edit form

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

Home > Fobi > My test form

Edit form

Elements
Handlers
Properties

Change form elements
Add form element +

(Content video):

Delusional Insanity - To Far Beyond



Edit Delete

Name (Text):

Edit Delete

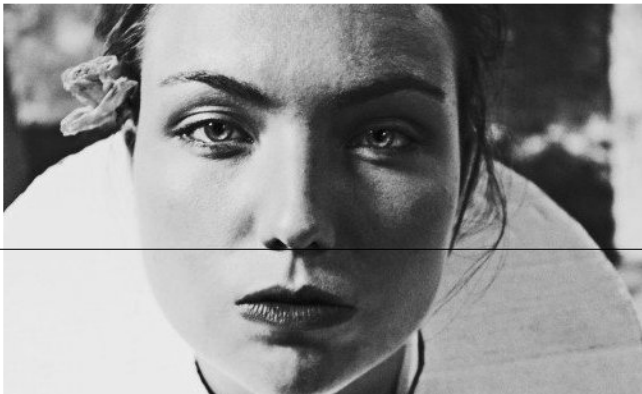
Description (Textarea):

Edit Delete

Website (URL):

Edit Delete

(Content image):



Edit form

Elements

Handlers

Properties

Change form elements

(Content video):

Delusional Insanity - To Far Beyond



Edit

Delete

Name (Text):

Edit

Delete

Description (Textarea):

Edit

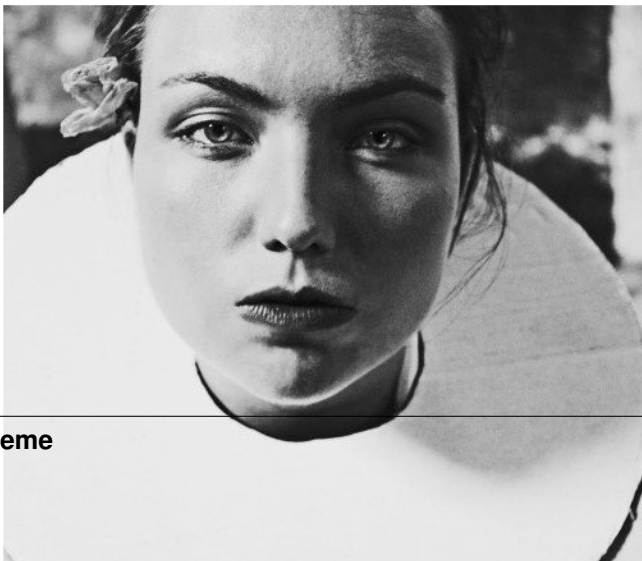
Delete

Website (URL):

Edit

Delete

(Content image):



Add form element +

Content

Content video

Content text

Content image

Fields

Select multiple

Textarea

Select model object

URL

Text

DateTime

Radio

File

Boolean

Date

Integer

Hidden

Password

Email

Select

Testing

Dummy

30.2. Simple theme

89

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

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

Add "Hidden" element to the form

Label:

Name:

Initial:

Max length:

Required: ☐

Add

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

Home > Fobi > My test form

Edit form

Elements Handlers Properties

Change form handlers

Add form handler +

Handler	Actions
DB store	Delete
Mail ?	Edit Delete

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

Home > Fobi > My test form

Edit form

Elements Handlers Properties

Change form properties

Name:

Public? ☒

Makes your form visible to the public.

Success page title:

Custom message title to display after valid form is submitted

Success page body:

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

Custom message text to display after valid form is submitted

Save

View form



Name:

Description:

Website:



Image: No file selected.

Agree? ☐

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

Contents:

31.1 fobi package

31.1.1 Subpackages

fobi.contrib package

Subpackages

fobi.contrib.apps package

Subpackages

fobi.contrib.apps.djangocms_integration package

Submodules

fobi.contrib.apps.djangocms_integration.apps module

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

fobi.contrib.apps.djangocms_integration.cms_plugins module

fobi.contrib.apps.djangocms_integration.conf module

```
fobi.contrib.apps.djangocms_integration.conf.get_setting(setting, override=None)
    Get a setting from fobi.contrib.apps.djangocms_integration conf module, falling back to the
    default.
```

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

Parameters

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

Returns Setting value.

fobi.contrib.apps.djangocms_integration.defaults module

fobi.contrib.apps.djangocms_integration.helpers module

fobi.contrib.apps.djangocms_integration.models module

fobi.contrib.apps.djangocms_integration.settings module

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

Module contents

fobi.contrib.apps.feincms_integration package

Submodules

fobi.contrib.apps.feincms_integration.apps module

class `fobi.contrib.apps.feincms_integration.apps.Config` (*app_name, app_module*)

Bases: `django.apps.config AppConfig`

label = 'fobi_contrib_apps_feincms_integration'

name = 'fobi.contrib.apps.feincms_integration'

fobi.contrib.apps.feincms_integration.conf module

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

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

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

Parameters

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

Returns Setting value.

fobi.contrib.apps.feincms_integration.defaults module

fobi.contrib.apps.feincms_integration.helpers module

fobi.contrib.apps.feincms_integration.settings module

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

fobi.contrib.apps.feincms_integration.widgets module

Module contents

fobi.contrib.apps.mezzanine_integration package

Submodules

fobi.contrib.apps.mezzanine_integration.admin module

fobi.contrib.apps.mezzanine_integration.apps module

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

fobi.contrib.apps.mezzanine_integration.conf module

```
fobi.contrib.apps.mezzanine_integration.conf.get_setting(setting, override=None)
    Get a setting from fobi.contrib.apps.mezzanine_integration conf module, falling back to the default.
```

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

Parameters

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

Returns Setting value.

fobi.contrib.apps.mezzanine_integration.defaults module

fobi.contrib.apps.mezzanine_integration.helpers module

fobi.contrib.apps.mezzanine_integration.models module

fobi.contrib.apps.mezzanine_integration.page_processors module

fobi.contrib.apps.mezzanine_integration.settings module

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

Module contents

fobi.contrib.apps.wagtail_integration package

Submodules

fobi.contrib.apps.wagtail_integration.apps module

fobi.contrib.apps.wagtail_integration.conf module

fobi.contrib.apps.wagtail_integration.defaults module

fobi.contrib.apps.wagtail_integration.helpers module

fobi.contrib.apps.wagtail_integration.models module

fobi.contrib.apps.wagtail_integration.settings module

Module contents

Module contents

fobi.contrib.plugins package

Subpackages

fobi.contrib.plugins.form_elements package

Subpackages

fobi.contrib.plugins.form_elements.content package

Subpackages

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

Submodules

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

class fobi.contrib.plugins.form_elements.content.content_image.apps.**Config** (*app_name*, *app_module*)

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_plugins_form_elements_content_content_image'

name = 'fobi.contrib.plugins.form_elements.content.content_image'

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

fobi.contrib.plugins.form_elements.content.content_image.conf.**get_setting** (*setting*, *override=None*)

Get a setting from fobi.contrib.plugins.form_elements.content.content_image.conf module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

fobi.contrib.plugins.form_elements.content.content_image.helpers.**handle_uploaded_file** (*image_file*)

Parameters *image_file* (django.core.files.uploadedfile.InMemoryUploadedFile) –

Return string Path to the image (relative).

fobi.contrib.plugins.form_elements.content.content_image.helpers.**get_crop_filter** (*fit_method*)

fobi.contrib.plugins.form_elements.content.content_image.helpers.**delete_file** (*image_file*)

Delete file from disc.

fobi.contrib.plugins.form_elements.content.content_image.helpers.**ensure_unique_filename** (*destination*)

Makes sure filenames are never overwritten.

Parameters *destination* (*string*) –

Return string

fobi.contrib.plugins.form_elements.content.content_image.helpers.**clone_file** (*source_filename*, *relative_path=True*)

Clones the file.

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

Return string Filename of the cloned file.

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

- FIT_METHOD_CROP_SMART (string)
- FIT_METHOD_CROP_CENTER (string)
- FIT_METHOD_CROP_SCALE (string)
- FIT_METHOD_FIT_WIDTH (string)
- FIT_METHOD_FIT_HEIGHT (string)
- DEFAULT_FIT_METHOD (string)
- FIT_METHODS_CHOICES (tuple)
- FIT_METHODS_CHOICES_WITH_EMPTY_OPTION (list)
- IMAGES_UPLOAD_DIR (string)

Module contents

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

Submodules

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

class fobi.contrib.plugins.form_elements.content.content_text.apps.**Config** (*app_name*, *app_module*)

Bases: django.apps.config AppConfig

label = 'fobi_contrib_plugins_form_elements_content_content_text'

name = 'fobi.contrib.plugins.form_elements.content.content_text'

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

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

Module contents

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

Submodules

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

class fobi.contrib.plugins.form_elements.content.content_video.apps.**Config** (*app_name*, *app_module*)

Bases: django.apps.config AppConfig

label = 'fobi_contrib_plugins_form_elements_content_content_video'

name = 'fobi.contrib.plugins.form_elements.content.content_video'

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

`fobi.contrib.plugins.form_elements.content.content_video.conf.get_setting(setting, override=None)`

Get a setting from `fobi.contrib.plugins.form_elements.content.content_video.conf` module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

Module contents

fobi.contrib.plugins.form_elements.fields package

Subpackages

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

Submodules

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

`class fobi.contrib.plugins.form_elements.fields.boolean.apps.Config(app_name, app_module)`

Bases: `django.apps.config AppConfig`

`label = 'fobi_contrib_plugins_form_elements_fields_boolean'`

`name = 'fobi.contrib.plugins.form_elements.fields.boolean'`

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

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

Module contents

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

Submodules

`fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps` module

`class fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps.Config` (*app_name, app_module*)

Bases: `django.apps.config AppConfig`

`label = 'fobi_contrib_plugins_form_elements_fields_checkbox_select_multiple'`

`name = 'fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple'`

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

`fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf.get_setting` (*setting, override=None*)

Get a setting from `fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple` conf module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.date.apps.Config(app_name,
                                                                app_module)

    Bases: django.apps.config.AppConfig

    label = 'fobi_contrib_plugins_form_elements_fields_date'
    name = 'fobi.contrib.plugins.form_elements.fields.date'
```

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

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.date_drop_down.apps.Config(app_name,
                                                                app_module)

    Bases: django.apps.config.AppConfig

    label = 'fobi_contrib_plugins_form_elements_fields_date_drop_down'
    name = 'fobi.contrib.plugins.form_elements.fields.date_drop_down'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.datetime.apps.Config(app_name,
                                                                app_module)

    Bases: django.apps.config.AppConfig

    label = 'fobi_contrib_plugins_form_elements_fields_datetime'
    name = 'fobi.contrib.plugins.form_elements.fields.datetime'
```

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

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.decimal.apps.Config(app_name,  
                                                                    app_module)
```

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

```
    label = 'fobi_contrib_plugins_form_elements_fields_decimal'
```

```
    name = 'fobi.contrib.plugins.form_elements.fields.decimal'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.email.apps.Config(app_name,  
                                                                    app_module)
```

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

```
    label = 'fobi_contrib_plugins_form_elements_fields_email'
```

```
    name = 'fobi.contrib.plugins.form_elements.fields.email'
```

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

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

Module contents

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

Submodules

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

class fobi.contrib.plugins.form_elements.fields.file.apps.**Config**(*app_name*,
app_module)

Bases: django.apps.config AppConfig

label = 'fobi_contrib_plugins_form_elements_fields_file'

name = 'fobi.contrib.plugins.form_elements.fields.file'

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

fobi.contrib.plugins.form_elements.fields.file.conf.**get_setting**(*setting*, *override=None*)

Get a setting from fobi.contrib.plugins.form_elements.fields.conf module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

- FILES_UPLOAD_DIR (string)

Module contents

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

Submodules

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

class fobi.contrib.plugins.form_elements.fields.float.apps.**Config**(*app_name*,
app_module)

Bases: django.apps.config AppConfig

label = 'fobi_contrib_plugins_form_elements_fields_float'

name = 'fobi.contrib.plugins.form_elements.fields.float'

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.hidden.apps.Config(app_name,  
                                                                    app_module)
```

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

```
    label = 'fobi_contrib_plugins_form_elements_fields_hidden'
```

```
    name = 'fobi.contrib.plugins.form_elements.fields.hidden'
```

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

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

Module contents

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

Submodules

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

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

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

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

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.input.apps.Config(app_name,  
                                                                app_module)
```

Bases: `django.apps.config.AppConfig`

label = 'fobi_contrib_plugins_form_elements_fields_input'

name = 'fobi.contrib.plugins.form_elements.fields.input'

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

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.integer.apps.Config(app_name,  
                                                                app_module)
```

Bases: `django.apps.config.AppConfig`

label = 'fobi_contrib_plugins_form_elements_fields_integer'

name = 'fobi.contrib.plugins.form_elements.fields.integer'

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config(app_name,  
                                                                    app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_ip_address'  
    name = 'fobi.contrib.plugins.form_elements.fields.ip_address'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config(app_name,  
                                                                    app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_null_boolean'  
    name = 'fobi.contrib.plugins.form_elements.fields.null_boolean'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.password.apps.Config(app_name,  
                                                                    app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_password'  
    name = 'fobi.contrib.plugins.form_elements.fields.password'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.radio.apps.Config(app_name,
                                                                app_module)

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_plugins_form_elements_fields_radio'
name = 'fobi.contrib.plugins.form_elements.fields.radio'
```

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

```
fobi.contrib.plugins.form_elements.fields.radio.conf.get_setting(setting, override=None)

Get a setting from fobi.contrib.plugins.form_elements.fields.radio conf module, falling back to the default.
If override is not None, it will be used instead of the setting.
```

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.regex.apps.Config(app_name,  
                                                                app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_regex'  
    name = 'fobi.contrib.plugins.form_elements.fields.regex'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.select.apps.Config(app_name,  
                                                                app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_select'  
    name = 'fobi.contrib.plugins.form_elements.fields.select'
```

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

```
fobi.contrib.plugins.form_elements.fields.select.conf.get_setting(setting,  
                                                                    over-  
                                                                    ride=None)  
  
    Get a setting from fobi.contrib.plugins.form_elements.fields.select conf module, falling back to the default.  
    If override is not None, it will be used instead of the setting.
```

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

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

Submodules

`fobi.contrib.plugins.form_elements.fields.select_model_object.apps` module

`class fobi.contrib.plugins.form_elements.fields.select_model_object.apps.Config` (*app_name*, *app_module*)

Bases: `django.apps.config AppConfig`

`label = 'fobi_contrib_plugins_form_elements_fields_select_model_object'`

`name = 'fobi.contrib.plugins.form_elements.fields.select_model_object'`

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

`fobi.contrib.plugins.form_elements.fields.select_model_object.conf.get_setting` (*setting*, *override=None*)

Get a setting from `fobi.contrib.plugins.form_elements.fields.select_model_object` conf module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

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

Submodules

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

class fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps.**Config**(*app_name*, *app_module*)

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_plugins_form_elements_fields_select_mptt_model_object'

name = 'fobi.contrib.plugins.form_elements.fields.select_mptt_model_object'

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

fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.conf.**get_setting**(*setting*, *override*=None)

Get a setting from *fobi.contrib.plugins.form_elements.fields.select_mptt_model_object* conf module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

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

Submodules

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

class fobi.contrib.plugins.form_elements.fields.select_multiple.apps.**Config**(*app_name*, *app_module*)

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_plugins_form_elements_fields_select_multiple'

name = 'fobi.contrib.plugins.form_elements.fields.select_multiple'

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

`fobi.contrib.plugins.form_elements.fields.select_multiple.conf.get_setting(setting, override=None)`

Get a setting from *fobi.contrib.plugins.form_elements.fields.select_multiple* conf module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

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

Submodules

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

`class fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps.Config(app)`

Bases: `django.apps.config AppConfig`

label = 'fobi_contrib_plugins_form_elements_fields_select_multiple_model_objects'

name = 'fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects'

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

`fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf.get_setting(setting, override=None)`

Get a setting from *fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects* conf module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

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

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

Module contents

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

Submodules

`fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps` module

`class fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps.Config`

Bases: `django.apps.config AppConfig`

`label = 'fobi_contrib_plugins_form_elements_fields_select_multiple_mptt_model_objects'`

`name = 'fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects'`

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

`fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf.get_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.

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

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

`fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.forms` module

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.slug.apps.Config(app_name,  
                                                                app_module)
```

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

```
    label = 'fobi_contrib_plugins_form_elements_fields_slug'
```

```
    name = 'fobi.contrib.plugins.form_elements.fields.slug'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.text.apps.Config(app_name,  
                                                                app_module)
```

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

```
    label = 'fobi_contrib_plugins_form_elements_fields_text'
```

```
    name = 'fobi.contrib.plugins.form_elements.fields.text'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.textarea.apps.Config(app_name,  
                                                                    app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_textarea'  
    name = 'fobi.contrib.plugins.form_elements.fields.textarea'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.time.apps.Config(app_name,  
                                                                    app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_time'  
    name = 'fobi.contrib.plugins.form_elements.fields.time'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.fields.url.apps.Config(app_name,  
                                                                    app_module)  
  
    Bases: django.apps.config.AppConfig  
  
    label = 'fobi_contrib_plugins_form_elements_fields_url'  
    name = 'fobi.contrib.plugins.form_elements.fields.url'
```

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

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

Module contents

Module contents

fobi.contrib.plugins.form_elements.security package

Subpackages

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.security.captcha.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    label = 'fobi_contrib_plugins_form_elements_security_captcha'
    name = 'fobi.contrib.plugins.form_elements.security.captcha'
```

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

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

Module contents

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

Submodules

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

```
class fobi.contrib.plugins.form_elements.security.honeypot.apps.Config(app_name,
                                                                    app_module)
    Bases: django.apps.config AppConfig
    label = 'fobi_contrib_plugins_form_elements_security_honeypot'
    name = 'fobi.contrib.plugins.form_elements.security.honeypot'
```

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

`fobi.contrib.plugins.form_elements.security.honeypot.conf.get_setting(setting, override=None)`

Get a setting from `fobi.contrib.plugins.form_elements.security.honeypot.conf` module, falling back to the default.

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

Parameters

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

Returns Setting value.

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

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

`class fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotField(max_length=None, min_length=None, strip=True, *args, **kwargs)`

Bases: `django.forms.fields.CharField`

clean (*value*)

Check that honeypot value remained the same.

default_error_messages = {'invalid': <django.utils.functional.__proxy__ object at 0x7f93925a5090>}

widget

alias of `HiddenInput`

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

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

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

- `HONEYPOT_VALUE` (string)

Module contents

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

Submodules

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

class fobi.contrib.plugins.form_elements.security.recaptcha.apps.**Config**(*app_name*,
app_module)

Bases: django.apps.config AppConfig

label = 'fobi_contrib_plugins_form_elements_security_recaptcha'

name = 'fobi.contrib.plugins.form_elements.security.recaptcha'

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

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

Module contents

Module contents

fobi.contrib.plugins.form_elements.test package

Subpackages

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

Submodules

fobi.contrib.plugins.form_elements.test.dummy.apps module

class fobi.contrib.plugins.form_elements.test.dummy.apps.**Config**(*app_name*,
app_module)

Bases: django.apps.config AppConfig

label = 'fobi_contrib_plugins_form_elements_test_dummy'

name = 'fobi.contrib.plugins.form_elements.test.dummy'

fobi.contrib.plugins.form_elements.test.dummy.fobi_form_elements module

fobi.contrib.plugins.form_elements.test.dummy.widgets module

Module contents

Module contents

Module contents

fobi.contrib.plugins.form_handlers package

Subpackages

fobi.contrib.plugins.form_handlers.db_store package

Submodules

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

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  
  
    label = 'fobi_contrib_plugins_form_handlers_db_store'  
    name = 'fobi.contrib.plugins.form_handlers.db_store'
```

fobi.contrib.plugins.form_handlers.db_store.conf module

```
fobi.contrib.plugins.form_handlers.db_store.conf.get_setting(setting,      over-  
                                                             ride=None)  
  
    Get a setting from fobi.contrib.plugins.form_handlers.db_store conf module, falling back  
    to the default.  
  
    If override is not None, it will be used instead of the setting.
```

Parameters

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

Returns Setting value.

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

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

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

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

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

- **CSV_DELIMITER** (string)
- **CSV_QUOTECHAR** (string)

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

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

fobi.contrib.plugins.form_handlers.db_store.widgets module

Module contents

fobi.contrib.plugins.form_handlers.http_repost package

Submodules

fobi.contrib.plugins.form_handlers.http_repost.apps module

```
class fobi.contrib.plugins.form_handlers.http_repost.apps.Config(app_name,
                                                                app_module)
    Bases: django.apps.config AppConfig
    label = 'fobi_contrib_plugins_form_handlers_http_repost'
    name = 'fobi.contrib.plugins.form_handlers.http_repost'
```

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

fobi.contrib.plugins.form_handlers.http_repost.forms module

Module contents

fobi.contrib.plugins.form_handlers.mail package

Submodules

fobi.contrib.plugins.form_handlers.mail.apps module

```
class fobi.contrib.plugins.form_handlers.mail.apps.Config(app_name, app_module)
    Bases: django.apps.config AppConfig
    label = 'fobi_contrib_plugins_form_handlers_mail'
    name = 'fobi.contrib.plugins.form_handlers.mail'
```

fobi.contrib.plugins.form_handlers.mail.conf module

```
fobi.contrib.plugins.form_handlers.mail.conf.get_setting(setting, override=None)
    Get a setting from fobi.contrib.plugins.form_handlers.mail conf module, falling back to the
    default.
```

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

Parameters

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

Returns Setting value.

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

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

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

Bases: django.forms.fields.Field

code = 'invalid'

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

to_python(value)
    Normalize data to a list of strings.

validate(value)
    Check if value consists only of valid emails.

widget
    alias of MultiEmailWidget
```

fobi.contrib.plugins.form_handlers.mail.fobi_form_handlers module

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

fobi.contrib.plugins.form_handlers.mail.helpers module

```
fobi.contrib.plugins.form_handlers.mail.helpers.send_mail(subject,          mes-
                                                         sage,
                                                         from_email,
                                                         recipient_list,
                                                         fail_silently=False,
                                                         auth_user=None,
                                                         auth_password=None,
                                                         connection=None,
                                                         html_message=None,
                                                         attachments=None)
```

Easy wrapper for sending a single message to a recipient list. All members of the recipient list will see the other recipients in the 'To' field.

If `auth_user` is `None`, the `EMAIL_HOST_USER` setting is used. If `auth_password` is `None`, the `EMAIL_HOST_PASSWORD` setting is used.

Note: The API for this method is frozen. New code wanting to extend the functionality should use the `EmailMessage` class directly.

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

fobi.contrib.plugins.form_handlers.mail.widgets module

```
class fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget (attrs=None)
    Bases: django.forms.widgets.Textarea

    is_hidden = False

    media

    prep_value (value)
        Prepare value before effectively render widget

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

Module contents

Module contents

fobi.contrib.plugins.form_importers package

Subpackages

fobi.contrib.plugins.form_importers.mailchimp_importer package

Submodules

fobi.contrib.plugins.form_importers.mailchimp_importer.apps module

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

fobi.contrib.plugins.form_importers.mailchimp_importer.tests module

Module contents

Module contents

Module contents

fobi.contrib.themes package

Subpackages

fobi.contrib.themes.bootstrap3 package

Subpackages

fobi.contrib.themes.bootstrap3.widgets package

Subpackages

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

Subpackages

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

Submodules

fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps module

```
class fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps.Config
```

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

```
    label = 'fobi_contrib_themes_bootstrap3_widgets_form_elements_date_bootstrap3_widget'
```

```
    name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget'
```

fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.fobi_form_elements module

Module contents

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

Submodules

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

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

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

```
    label = 'fobi_contrib_themes_bootstrap3_widgets_form_elements_datetime_bootstrap3_widget'
```

```
    name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget'
```

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

Module contents

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

Submodules

fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps module

class `fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps.Config`

Bases: `django.apps.config.AppConfig`

label = 'fobi_contrib_themes_bootstrap3_widgets_form_elements_dummy_bootstrap3_widget'

name = 'fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget'

fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.fobi_form_elements module

Module contents

Module contents

Module contents

Submodules

fobi.contrib.themes.bootstrap3.apps module

class `fobi.contrib.themes.bootstrap3.apps.Config(app_name, app_module)`

Bases: `django.apps.config.AppConfig`

label = 'fobi_contrib_themes_bootstrap3'

name = 'fobi.contrib.themes.bootstrap3'

fobi.contrib.themes.bootstrap3.fobi_themes module

Module contents

fobi.contrib.themes.djangocms_admin_style_theme package

Subpackages

fobi.contrib.themes.djangocms_admin_style_theme.widgets package

Subpackages

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

Subpackages

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

Submodules

fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps module

class fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps.**Config**

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_themes.djangocms_admin_style_theme_widgets_form_handlers_db_store'

name = 'fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store'

fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.fobi_form_elements module

Module contents

Module contents

Module contents

Submodules

fobi.contrib.themes.djangocms_admin_style_theme.apps module

class fobi.contrib.themes.djangocms_admin_style_theme.apps.**Config**(*app_name*,
app_module)

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_themes.djangocms_admin_style_theme'

name = 'fobi.contrib.themes.djangocms_admin_style_theme'

fobi.contrib.themes.djangocms_admin_style_theme.fobi_themes module

Module contents

fobi.contrib.themes.foundation5 package

Subpackages

fobi.contrib.themes.foundation5.widgets package

Subpackages

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

Subpackages

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

Submodules

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

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

Bases: `django.apps.config AppConfig`

label = 'fobi_contrib_themes_foundation5_widgets_form_elements_date_foundation5_widget'

name = 'fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget'

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

Module contents

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

Submodules

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

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

Bases: `django.apps.config AppConfig`

label = 'fobi_contrib_themes_foundation5_widgets_form_elements_datetime_foundation5_widget'

name = 'fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget'

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

Module contents

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

Submodules

fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps module

class fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps.Con

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_themes_foundation5_widgets_form_elements_dummy_foundation5_widget'

name = 'fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget'

fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.fobi_form_elements module

Module contents

Module contents

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

Subpackages

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

Submodules

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

class fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps.

Bases: django.apps.config.AppConfig

label = 'fobi_contrib_themes_foundation5_widgets_form_handlers_db_store_foundation5_widget'

name = 'fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget'

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

Module contents

Module contents

Module contents

Submodules

fobi.contrib.themes.foundation5.apps module

```
class fobi.contrib.themes.foundation5.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig

    label = 'fobi_contrib_themes_foundation5'
    name = 'fobi.contrib.themes.foundation5'
```

fobi.contrib.themes.foundation5.fobi_themes module

Module contents

fobi.contrib.themes.simple package

Subpackages

fobi.contrib.themes.simple.widgets package

Subpackages

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

Subpackages

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

Submodules

fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps module

```
class fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps.Config(app_name,
                                                                              app_module)
    Bases: django.apps.config.AppConfig

    label = 'fobi_contrib_themes_simple_widgets_form_handlers_db_store'
    name = 'fobi.contrib.themes.simple.widgets.form_handlers.db_store'
```

fobi.contrib.themes.simple.widgets.form_handlers.db_store.fobi_form_elements module

Module contents

Module contents

Module contents

Submodules

fobi.contrib.themes.simple.apps module

```
class fobi.contrib.themes.simple.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig

    label = 'fobi_contrib_themes_simple'
    name = 'fobi.contrib.themes.simple'
```

fobi.contrib.themes.simple.fobi_themes module

Module contents

Module contents

Module contents

fobi.integration package

Submodules

fobi.integration.helpers module

fobi.integration.processors module

Module contents

fobi.management package

Subpackages

fobi.management.commands package

Submodules

fobi.management.commands.fobi_find_broken_entries module

fobi.management.commands.fobi_migrate_03_to_04 module

fobi.management.commands.fobi_sync_plugins module

fobi.management.commands.fobi_update_plugin_data module

Module contents

Module contents

fobi.templatetags package

Submodules

fobi.templatetags.fobi_tags module

`fobi.templatetags.fobi_tags.get_fobi_plugin(parser, token)`

Gets the plugin. Note, that entry shall be a instance of `fobi.models.FormElementEntry` or `fobi.models.FormHandlerEntry`.

Syntax `{% get_fobi_plugin entry as [context_var_name] %}`

Example `{% get_fobi_plugin entry as plugin %}`

`{% get_fobi_plugin entry as plugin %} {{ plugin.render }}`

`fobi.templatetags.fobi_tags.get_fobi_form_handler_plugin_custom_actions(parser, token)`

Gets the form handler plugin custom actions. Note, that plugin shall be a instance of `fobi.models.FormHandlerEntry`.

Syntax `{% get_fobi_form_handler_plugin_custom_actions [plugin] [form_entry] as [context_var_name] %}`

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

`fobi.templatetags.fobi_tags.get_form_field_type(parser, token)`

Get form field type.

Syntax:

```
{% get_form_field_type [field] as [context_var_name] %}
```

Example:

```
{% get_form_field_type form.field as form_field_type %}
{% if form_field_type.is_checkbox %}
...
{% endif %}
```

`fobi.templatetags.fobi_tags.get_form_hidden_fields_errors(parser, token)`

Get form hidden fields errors.

Syntax `{% get_form_hidden_fields_errors [form] as [context_var_name] %}`

Example `{% get_form_hidden_fields_errors form as form_hidden_fields_errors %} {{ form_hidden_fields_errors.as_ul }}`

`fobi.templatetags.fobi_tags.has_edit_form_entry_permissions(parser, token)`

Checks the permissions

Syntax `{% has_edit_form_entry_permissions as [var_name] %}`

Example `{% has_edit_form_entry_permissions %}`

or

`{% has_edit_form_entry_permissions as has_permissions %}`

`fobi.templatetags.fobi_tags.render_auth_link(context)`
Render auth link.

`fobi.templatetags.future_compat` module

`fobi.templatetags.future_compat.firstof(parser, token, escape=False)`
Outputs the first variable passed that is not False, without escaping.

Outputs nothing if all the passed variables are False.

Sample usage:

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

This is equivalent to:

```
{% if var1 %}
    {{ var1|safe }}
{% elif var2 %}
    {{ var2|safe }}
{% elif var3 %}
    {{ var3|safe }}
{% endif %}
```

but obviously much cleaner!

You can also use a literal string as a fallback value in case all passed variables are False:

```
{% firstof var1 var2 var3 "fallback value" %}
```

If you want to escape the output, use a filter tag:

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

Module contents

fobi.tests package

Submodules

fobi.tests.base module

fobi.tests.constants module

fobi.tests.data module

fobi.tests.helpers module

fobi.tests.test_browser_build_dynamic_forms module

fobi.tests.test_core module

fobi.tests.test_dynamic_forms module

fobi.tests.test_sortable_dict module

Module contents

fobi.urls package

Submodules

fobi.urls.edit module

fobi.urls.view module

Module contents

31.1.2 Submodules

31.1.3 fobi.admin module

31.1.4 fobi.app module

`fobi.app.app_name` (*path*, *reduce_depth_by=1*)
Returns 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'*)

Parameters

- **path** (*str*) – Absolute app path (from project root).
- **config_app_path** (*str*) – Relative config path (from app root)

Return *str*

31.1.5 fobi.apps module

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

31.1.6 fobi.base module

31.1.7 fobi.compat module

31.1.8 fobi.conf module

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

31.1.9 fobi.constants module

31.1.10 fobi.context_processors module

31.1.11 fobi.data_structures module

```
class fobi.data_structures.SortableDict(data=None)
    Bases: dict
```

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 ()

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*)

Inserts 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*)

Inserts the {key: value} before the *target_key*.

Parameters

- **target_key** (*immutable*) –
- **key** (*immutable*) –
- **value** (*mutable*) –
- **fail_silently** (*boolean*) –
- **offset** (*int*) –

Return bool

items ()

iteritems ()

iterkeys ()

intervalues ()

keys ()

move_after_key (*source_key*, *target_key*, *fail_silently=True*)

Moves 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*)

Moves the {key: value} before the given *source_key*.

Parameters

- **source_key** (*immutable*) –
- **target_key** (*immutable*) –
- **fail_silently** (*boolean*) –
- **offset** (*int*) –

Return bool

```
pop(k, *args)
popitem()
setdefault(key, default)
update(dict_)
value_for_index(index)
    Returns the value of the item at the given zero-based index.
values()
```

31.1.12 fobi.decorators module

`fobi.decorators.permissions_required(perms, satisfy='all', login_url=None, raise_exception=False)`

Checks 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)`

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)`

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
```

31.1.13 fobi.defaults module

31.1.14 fobi.discover module

31.1.15 fobi.dynamic module

`fobi.dynamic.assemble_form_class(form_entry, base_class=<class 'django.forms.forms.BaseForm'>, request=None, origin=None, origin_kwargs_update_func=None, origin_return_func=None, form_element_entries=None)`

Assembles 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).

31.1.16 fobi.exceptions module

exception `fobi.exceptions.BaseException`

Bases: `exceptions.Exception`

Base exception.

exception `fobi.exceptions.ImproperlyConfigured`

Bases: `fobi.exceptions.BaseException`

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

exception `fobi.exceptions.InvalidRegistryItemType`

Bases: `exceptions.ValueError`, `fobi.exceptions.BaseException`

Raised when an attempt is made to register an item in the registry which does not have a proper type.

exception `fobi.exceptions.DoesNotExist`

Bases: `fobi.exceptions.BaseException`

Raised when something does not exist.

exception `fobi.exceptions.ThemeDoesNotExist`

Bases: `fobi.exceptions.DoesNotExist`

Raised when no theme with given uid can be found.

exception `fobi.exceptions.PluginDoesNotExist`

Bases: `fobi.exceptions.DoesNotExist`

Raised when no plugin with given uid can be found.

exception `fobi.exceptions.FormElementPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

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

exception `fobi.exceptions.FormHandlerPluginDoesNotExist`

Bases: `fobi.exceptions.PluginDoesNotExist`

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

exception `fobi.exceptions.NoDefaultThemeSet`

Bases: `fobi.exceptions.ImproperlyConfigured`

Raised when no active theme is chosen.

exception `fobi.exceptions.FormPluginError`

Bases: `fobi.exceptions.BaseException`

Base error for form elements and handlers.

exception `fobi.exceptions.FormElementPluginError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form element plugin error occurs.

exception `fobi.exceptions.FormHandlerPluginError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form handler plugin error occurs.

exception `fobi.exceptions.FormCallbackError`

Bases: `fobi.exceptions.FormPluginError`

Raised when form callback error occurs.

31.1.17 fobi.form_importers module

31.1.18 fobi.form_utils module

31.1.19 fobi.forms module

31.1.20 fobi.helpers module

31.1.21 fobi.models module

31.1.22 fobi.settings module

- *RESTRICT_PLUGIN_ACCESS* (bool): If set to True, (Django) permission system for fobi plugins is enabled.
- *FORM_ELEMENT_PLUGINS_MODULE_NAME* (str): Name of the module to placed in the (external) apps in which the fobi form element plugin code should be implemented and registered.
- *FORM_HANDLER_PLUGINS_MODULE_NAME* (str): Name of the module to placed in the (external) apps in which the fobi form handler plugin code should be implemented and registered.
- *FORM_CALLBACKS_MODULE_NAME* (str): Name of the module to placed in the (external) apps in which the fobi form callback code should be implemented and registered.
- *FORM_HANDLER_PLUGINS_EXECUTION_ORDER* (tuple): Order in which the form handler plugins are to be executed.
- *DEBUG*

31.1.23 fobi.test module

31.1.24 fobi.utils module

31.1.25 fobi.validators module

`fobi.validators.url_exists(url, local=False)`
Check if URL exists.

Parameters

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

Return bool

31.1.26 fobi.views module

31.1.27 fobi.widgets module

```
class fobi.widgets.NumberInput(attrs=None)
    Bases: django.forms.widgets.TextInput
    input_type = u'number'
    media
```

```
class fobi.widgets.BooleanRadioSelect(*args, **kwargs):
    Bases: django.forms.widgets.RadioSelect

    Boolean radio select for Django.
```

Example

```
>>> class DummyForm(forms.Form):
>>>     agree = forms.BooleanField(label=_("Agree?"), required=False, widget=BooleanRadioSelect)
```

media

31.1.28 Module contents

31.2 Quick start

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

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

31.2.1 Part 1: standard Django installation

Example project code available [here](#).

Installation and configuration

Install the package in your environment.

```
pip install django-fobi
```

INSTALLED_APPS

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

1. The core.

```
'fobi',
```

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

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

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

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

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

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

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

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

Putting all together, you would have something like this.

```
INSTALLED_APPS = (
    # Used by fobi
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django.contrib.admin',

    # ...
    # Core
    'fobi',

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

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

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

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

# ...
)
```

TEMPLATE_CONTEXT_PROCESSORS

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

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


urlpatterns

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

```
urlpatterns = [
    # ...

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

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

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

    # ...

]
```

Update the database

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

```
$ ./manage.py syncdb
$ ./manage.py migrate
```

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

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

Specify the active theme

Specify the default theme in your *settings* module.

```
FOBI_DEFAULT_THEME = 'bootstrap3'
```

Permissions

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

```
FOBI_RESTRICT_PLUGIN_ACCESS = False
```

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

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

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

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

31.2.2 Part 2: Integration with DjangoCMS

Coming soon...

Indices and tables

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

- [1.1] Dashboard
- [1.2] Create a form
- [1.3] Edit form - form elements tab active, no elements yet
- [1.4] Edit form - form elements tab active, add a form element menu
- [1.5] Edit form - add a form element (URL plugin)
- [1.6] Edit form - form elements tab active, with form elements
- [1.7] Edit form - form handlers tab active, no handlers yet
- [1.8] Edit form - form handlers tab active, add form handler menu
- [1.9] Edit form - add a form handler (Mail plugin)
- [1.10] Edit form - form handlers tab active, with form handlers
- [1.11] Edit form - form properties tab active
- [1.12] View form
- [1.13] View form - form submitted (thanks page)
- [1.14] Edit form - add a form element (Video plugin)
- [1.15] Edit form - add a form element (Boolean plugin)
- [1.16] Edit form
- [1.17] View form
- [2.1] Edit form - form elements tab active, with form elements
- [2.2] Edit form - form elements tab active, add a form element menu
- [2.3] Edit form - add a form element (Hidden plugin)
- [2.4] Edit form - form handlers tab active, with form handlers
- [2.5] Edit form - form properties tab active
- [2.6] View form

f

fobi, 138	fobi.contrib.plugins.form_elements.content,
fobi.app, 131	99
fobi.apps, 132	fobi.contrib.plugins.form_elements.content.content,
fobi.conf, 132	98
fobi.constants, 132	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib, 128	97
fobi.contrib.apps, 96	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.djangocms_integration,	97
94	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.djangocms_integration.apps,	97
93	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.djangocms_integration.conf,	98
93	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.djangocms_integration.defaults,	98
94	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.djangocms_integration.settings,	98
94	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.feincms_integration,	99
95	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.feincms_integration.apps,	98
94	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.feincms_integration.conf,	99
94	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.feincms_integration.defaults,	99
94	fobi.contrib.plugins.form_elements.content.content,
fobi.contrib.apps.feincms_integration.settings,	99
95	fobi.contrib.plugins.form_elements.fields,
fobi.contrib.apps.mezzanine_integration,	115
96	fobi.contrib.plugins.form_elements.fields.boolean,
fobi.contrib.apps.mezzanine_integration.apps,	100
95	fobi.contrib.plugins.form_elements.fields.boolean.
fobi.contrib.apps.mezzanine_integration.conf,	99
95	fobi.contrib.plugins.form_elements.fields.checkbox,
fobi.contrib.apps.mezzanine_integration.defaults,	100
95	fobi.contrib.plugins.form_elements.fields.checkbox,
fobi.contrib.apps.mezzanine_integration.settings,	100
95	fobi.contrib.plugins.form_elements.fields.checkbox,
fobi.contrib.plugins, 121	100
fobi.contrib.plugins.form_elements, 117	100

```

fobi.contrib.plugins.form_elements.fields.checkbox.fields.null_boolean
100 106
fobi.contrib.plugins.form_elements.fields.date.contrib.plugins.form_elements.fields.null_boolean
101 106
fobi.contrib.plugins.form_elements.fields.date.apps.plugins.form_elements.fields.password
101 107
fobi.contrib.plugins.form_elements.fields.date.contrib.plugins.form_elements.fields.password
101 106
fobi.contrib.plugins.form_elements.fields.date.contrib.plugins.form_elements.fields.radio
101 107
fobi.contrib.plugins.form_elements.fields.date.contrib.plugins.form_elements.fields.radio.app
102 107
fobi.contrib.plugins.form_elements.fields.date.contrib.plugins.form_elements.fields.radio.con
101 107
fobi.contrib.plugins.form_elements.fields.decimal.contrib.plugins.form_elements.fields.radio.de
102 107
fobi.contrib.plugins.form_elements.fields.decimal.apps.plugins.form_elements.fields.regex
102 108
fobi.contrib.plugins.form_elements.fields.email.contrib.plugins.form_elements.fields.regex.app
102 108
fobi.contrib.plugins.form_elements.fields.email.apps.plugins.form_elements.fields.select
102 109
fobi.contrib.plugins.form_elements.fields.file.contrib.plugins.form_elements.fields.select.ap
103 108
fobi.contrib.plugins.form_elements.fields.file.apps.plugins.form_elements.fields.select.co
103 108
fobi.contrib.plugins.form_elements.fields.file.contrib.plugins.form_elements.fields.select.de
103 108
fobi.contrib.plugins.form_elements.fields.file.default.plugins.form_elements.fields.select_mo
103 109
fobi.contrib.plugins.form_elements.fields.file.settings.plugins.form_elements.fields.select_mo
103 109
fobi.contrib.plugins.form_elements.fields.file.contrib.plugins.form_elements.fields.select_mo
104 109
fobi.contrib.plugins.form_elements.fields.file.apps.plugins.form_elements.fields.select_mo
103 109
fobi.contrib.plugins.form_elements.fields.hidden.contrib.plugins.form_elements.fields.select_mp
104 110
fobi.contrib.plugins.form_elements.fields.hidden.apps.plugins.form_elements.fields.select_mp
104 110
fobi.contrib.plugins.form_elements.fields.input.contrib.plugins.form_elements.fields.select_mp
105 110
fobi.contrib.plugins.form_elements.fields.input.apps.plugins.form_elements.fields.select_mp
105 110
fobi.contrib.plugins.form_elements.fields.input.constant.plugins.form_elements.fields.select_mu
105 111
fobi.contrib.plugins.form_elements.fields.integer.contrib.plugins.form_elements.fields.select_mu
105 110
fobi.contrib.plugins.form_elements.fields.integer.apps.plugins.form_elements.fields.select_mu
105 111
fobi.contrib.plugins.form_elements.fields.ip_address.plugins.form_elements.fields.select_mu
106 111
fobi.contrib.plugins.form_elements.fields.ip_address.apps.plugins.form_elements.fields.select_mu
106 112

```


fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf,	111
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults,	112
fobi.contrib.plugins.form_elements.fields.slug,	113
fobi.contrib.plugins.form_elements.fields.text,	113
fobi.contrib.plugins.form_elements.fields.url,	115
fobi.contrib.plugins.form_elements.security.capture_tcha,	117
fobi.contrib.plugins.form_elements.security.honeypot.conf,	116
fobi.contrib.plugins.form_elements.security.honeypot.defaults,	116
fobi.contrib.plugins.form_elements.security.honeypot.fields,	116
fobi.contrib.plugins.form_elements.security.honeypot.settings,	116
fobi.contrib.plugins.form_elements.security.recaptcha,	117
fobi.contrib.plugins.form_handlers.db_store,	119
fobi.contrib.plugins.form_handlers.mail,	121
fobi.contrib.plugins.form_handlers.mail.apps,	121
fobi.contrib.plugins.form_handlers.mail.conf,	121
fobi.contrib.plugins.form_handlers.mail.defaults,	121
fobi.contrib.plugins.form_handlers.mail.fields,	121
fobi.contrib.plugins.form_handlers.mail.helpers,	120
fobi.contrib.plugins.form_handlers.mail.settings,	121
fobi.contrib.plugins.form_handlers.mail.widgets,	121
fobi.contrib.plugins.form_importers,	121
fobi.contrib.themes.bootstrap3,	123
fobi.contrib.themes.bootstrap3.apps,	123
fobi.contrib.themes.bootstrap3.widgets,	123

```

fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps,
122 fobi.dynamic, 135
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget,
123 fobi.integration, 128
fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps,
123 fobi.management.commands, 128
fobi.contrib.themes.djangocms_admin_style_theme.settings, 137
124 fobi.templatetags, 131
fobi.contrib.themes.djangocms_admin_style_theme.templatetags.fobi_tags, 129
124 fobi.templatetags.future_compat, 130
fobi.contrib.themes.djangocms_admin_style_theme.widgets,
124 fobi.validators, 137
fobi.contrib.themes.djangocms_admin_style_theme.widget.get37.form_handlers,
124
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store,
124
fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps,
124
fobi.contrib.themes.foundation5, 127
fobi.contrib.themes.foundation5.apps,
127
fobi.contrib.themes.foundation5.widgets,
126
fobi.contrib.themes.foundation5.widgets.form_elements,
126
fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget,
125
fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps,
125
fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget,
125
fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps,
125
fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget,
126
fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps,
126
fobi.contrib.themes.foundation5.widgets.form_handlers,
126
fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget,
126
fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps,
126
fobi.contrib.themes.simple, 128
fobi.contrib.themes.simple.apps, 128
fobi.contrib.themes.simple.widgets, 127
fobi.contrib.themes.simple.widgets.form_handlers,
127
fobi.contrib.themes.simple.widgets.form_handlers.db_store,
127
fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps,
127
fobi.data_structures, 132
fobi.decorators, 134

```

A

all_permissions_required() (in module fobi.decorators),
134
any_permission_required() (in module fobi.decorators),
134
app_config() (in module fobi.app), 131
app_name() (in module fobi.app), 131
assemble_form_class() (in module fobi.dynamic), 135

B

BaseException, 135
BooleanRadioSelect (class in fobi.widgets), 137

C

clean() (fobi.contrib.plugins.form_elements.security.honeypot.fields.HoneypotField
method), 116
clear() (fobi.data_structures.SortableDict method), 132
clone_file() (in module fobi.contrib.plugins.form_elements.content.content_image.helpers),
97
code (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField
attribute), 120
Config (class in fobi.apps), 132
Config (class in fobi.contrib.apps.djangocms_integration.apps),
93
Config (class in fobi.contrib.apps.feincms_integration.apps),
94
Config (class in fobi.contrib.apps.mezzanine_integration.apps),
95
Config (class in fobi.contrib.plugins.form_elements.content.content_image.apps),
97
Config (class in fobi.contrib.plugins.form_elements.content.content_text.apps),
98
Config (class in fobi.contrib.plugins.form_elements.content.content_video.apps),
98
Config (class in fobi.contrib.plugins.form_elements.fields.boolean.apps),
99
Config (class in fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps),
100
Config (class in fobi.contrib.plugins.form_elements.fields.date.apps),
101
Config (class in fobi.contrib.plugins.form_elements.fields.date_drop_down.apps),
101
Config (class in fobi.contrib.plugins.form_elements.fields.datetime.apps),
101
Config (class in fobi.contrib.plugins.form_elements.fields.decimal.apps),
102
Config (class in fobi.contrib.plugins.form_elements.fields.email.apps),
102
Config (class in fobi.contrib.plugins.form_elements.fields.file.apps),
103
Config (class in fobi.contrib.plugins.form_elements.fields.float.apps),
103
Config (class in fobi.contrib.plugins.form_elements.fields.hidden.apps),
104
Config (class in fobi.contrib.plugins.form_elements.fields.input.apps),
105
Config (class in fobi.contrib.plugins.form_elements.fields.integer.apps),
105
Config (class in fobi.contrib.plugins.form_elements.fields.ip_address.apps),
106
Config (class in fobi.contrib.plugins.form_elements.fields.null_boolean.apps),
106
Config (class in fobi.contrib.plugins.form_elements.fields.password.apps),
106
Config (class in fobi.contrib.plugins.form_elements.fields.radio.apps),
107
Config (class in fobi.contrib.plugins.form_elements.fields.regex.apps),
108
Config (class in fobi.contrib.plugins.form_elements.fields.select.apps),
108
Config (class in fobi.contrib.plugins.form_elements.fields.select_model_obj.apps),
109
Config (class in fobi.contrib.plugins.form_elements.fields.select_mptt_model_obj.apps),
110
Config (class in fobi.contrib.plugins.form_elements.fields.select_multiple.apps),
110
Config (class in fobi.contrib.plugins.form_elements.fields.select_multiple_model_obj.apps),
111

Config (class in fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps),
 112
 Config (class in fobi.contrib.plugins.form_elements.fields.slug.apps), default_error_messages (fobi.contrib.plugins.form_elements.security.honeypot.apps),
 113 attribute), 116
 delete_file() (in module
 Config (class in fobi.contrib.plugins.form_elements.fields.text.apps), fobi.contrib.plugins.form_elements.content.content_image.helper
 113 97
 Config (class in fobi.contrib.plugins.form_elements.fields.textarea.apps), DoesNotExist, 136
 114
 Config (class in fobi.contrib.plugins.form_elements.fields.time.apps), E
 114 ensure_unique_filename() (in module
 Config (class in fobi.contrib.plugins.form_elements.fields.url.apps), fobi.contrib.plugins.form_elements.content.content_image.helper
 114 97
 Config (class in fobi.contrib.plugins.form_elements.security.captcha.apps), F
 115
 Config (class in fobi.contrib.plugins.form_elements.security.honeypot.apps), firstof (in module fobi.templatetags.future_compat),
 115 130
 Config (class in fobi.contrib.plugins.form_elements.security.recaptcha.apps), fobi (module), 138
 117 fobi.app (module), 131
 Config (class in fobi.contrib.plugins.form_elements.test.dummy.apps), fobi.apps (module), 132
 117 fobi.conf (module), 132
 Config (class in fobi.contrib.plugins.form_handlers.db_store.apps), fobi.constants (module), 132
 118 fobi.contrib (module), 128
 Config (class in fobi.contrib.plugins.form_handlers.http_repost.apps), fobi.contrib.apps (module), 96
 119 fobi.contrib.apps.djangocms_integration (module), 94
 Config (class in fobi.contrib.plugins.form_handlers.mail.apps), fobi.contrib.apps.djangocms_integration.apps (module),
 119 93
 Config (class in fobi.contrib.themes.bootstrap3.apps), 123 fobi.contrib.apps.djangocms_integration.conf (module),
 Config (class in fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3_widget.apps), (module),
 122 93
 Config (class in fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3_widget.apps), (mod-
 122 ule), 94
 Config (class in fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3_widget.apps), (mod-
 123 ule), 94
 Config (class in fobi.contrib.themes.djangocms_admin_style_theme.apps), fobi.contrib.apps.feincms_integration (module), 95
 124 fobi.contrib.apps.feincms_integration.apps (module), 94
 Config (class in fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps), fobi.contrib.apps.feincms_integration.conf (module), 94
 124 fobi.contrib.apps.feincms_integration.defaults (module),
 94
 Config (class in fobi.contrib.themes.foundation5.apps), fobi.contrib.apps.feincms_integration.settings (module),
 127 95
 Config (class in fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5_widget.apps), fobi.contrib.apps.mezzanine_integration (module), 96
 125 fobi.contrib.apps.mezzanine_integration.apps (module),
 Config (class in fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps), (module),
 125 95
 Config (class in fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps), fobi.contrib.apps.mezzanine_integration.conf (module),
 126 95
 Config (class in fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps), fobi.contrib.apps.mezzanine_integration.defaults (mod-
 126 ule), 95
 Config (class in fobi.contrib.themes.simple.apps), 128 fobi.contrib.apps.mezzanine_integration.settings (mod-
 Config (class in fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps), (ule), 95
 127 fobi.contrib.plugins (module), 121
 copy() (fobi.data_structures.SortableDict method), 132 fobi.contrib.plugins.form_elements (module), 117
 fobi.contrib.plugins.form_elements.content (module), 99
 fobi.contrib.plugins.form_elements.content.content_image
 (module), 98

[fobi.contrib.plugins.form_elements.content.content_image.apps \(module\), 97](#)
[fobi.contrib.plugins.form_elements.content.content_image.conf \(module\), 97](#)
[fobi.contrib.plugins.form_elements.content.content_image.defaults \(module\), 97](#)
[fobi.contrib.plugins.form_elements.content.content_image.fields \(module\), 97](#)
[fobi.contrib.plugins.form_elements.content.content_image.fields.apps \(module\), 97](#)
[fobi.contrib.plugins.form_elements.content.content_image.fields.conf \(module\), 98](#)
[fobi.contrib.plugins.form_elements.content.content_text \(module\), 98](#)
[fobi.contrib.plugins.form_elements.content.content_text.apps \(module\), 98](#)
[fobi.contrib.plugins.form_elements.content.content_video \(module\), 99](#)
[fobi.contrib.plugins.form_elements.content.content_video.apps \(module\), 98](#)
[fobi.contrib.plugins.form_elements.content.content_video.conf \(module\), 99](#)
[fobi.contrib.plugins.form_elements.content.content_video.defaults \(module\), 99](#)
[fobi.contrib.plugins.form_elements.content.content_video.fields \(module\), 99](#)
[fobi.contrib.plugins.form_elements.fields \(module\), 115](#)
[fobi.contrib.plugins.form_elements.fields.boolean \(module\), 100](#)
[fobi.contrib.plugins.form_elements.fields.boolean.apps \(module\), 99](#)
[fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple \(module\), 100](#)
[fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps \(module\), 100](#)
[fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.conf \(module\), 100](#)
[fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.defaults \(module\), 100](#)
[fobi.contrib.plugins.form_elements.fields.date \(module\), 101](#)
[fobi.contrib.plugins.form_elements.fields.date.apps \(module\), 101](#)
[fobi.contrib.plugins.form_elements.fields.date_drop_down \(module\), 101](#)
[fobi.contrib.plugins.form_elements.fields.date_drop_down.apps \(module\), 101](#)
[fobi.contrib.plugins.form_elements.fields.datetime \(module\), 102](#)
[fobi.contrib.plugins.form_elements.fields.datetime.apps \(module\), 101](#)
[fobi.contrib.plugins.form_elements.fields.decimal \(module\), 102](#)
[fobi.contrib.plugins.form_elements.fields.decimal.apps \(module\), 102](#)
[fobi.contrib.plugins.form_elements.fields.email \(module\), 102](#)
[fobi.contrib.plugins.form_elements.fields.email.apps \(module\), 102](#)
[fobi.contrib.plugins.form_elements.fields.file \(module\), 103](#)
[fobi.contrib.plugins.form_elements.fields.file.apps \(module\), 103](#)
[fobi.contrib.plugins.form_elements.fields.file.conf \(module\), 103](#)
[fobi.contrib.plugins.form_elements.fields.file.defaults \(module\), 103](#)
[fobi.contrib.plugins.form_elements.fields.file.settings \(module\), 103](#)
[fobi.contrib.plugins.form_elements.fields.float \(module\), 104](#)
[fobi.contrib.plugins.form_elements.fields.float.apps \(module\), 103](#)
[fobi.contrib.plugins.form_elements.fields.hidden \(module\), 104](#)
[fobi.contrib.plugins.form_elements.fields.hidden.apps \(module\), 104](#)
[fobi.contrib.plugins.form_elements.fields.input \(module\), 105](#)
[fobi.contrib.plugins.form_elements.fields.input.apps \(module\), 105](#)
[fobi.contrib.plugins.form_elements.fields.input.constants \(module\), 105](#)
[fobi.contrib.plugins.form_elements.fields.integer \(module\), 105](#)
[fobi.contrib.plugins.form_elements.fields.integer.apps \(module\), 105](#)
[fobi.contrib.plugins.form_elements.fields.ip_address \(module\), 106](#)
[fobi.contrib.plugins.form_elements.fields.ip_address.apps \(module\), 106](#)
[fobi.contrib.plugins.form_elements.fields.null_boolean \(module\), 106](#)
[fobi.contrib.plugins.form_elements.fields.null_boolean.apps \(module\), 106](#)
[fobi.contrib.plugins.form_elements.fields.password \(module\), 107](#)
[fobi.contrib.plugins.form_elements.fields.password.apps \(module\), 106](#)
[fobi.contrib.plugins.form_elements.fields.radio \(module\), 107](#)
[fobi.contrib.plugins.form_elements.fields.radio.apps \(module\), 107](#)
[fobi.contrib.plugins.form_elements.fields.radio.conf \(module\), 107](#)
[fobi.contrib.plugins.form_elements.fields.radio.defaults \(module\), 107](#)
[fobi.contrib.plugins.form_elements.fields.regex \(module\), 108](#)

fobi.contrib.plugins.form_elements.fields.regex.apps (module), 108	fobi.contrib.plugins.form_elements.fields.text (module), 113
fobi.contrib.plugins.form_elements.fields.select (module), 109	fobi.contrib.plugins.form_elements.fields.text.apps (module), 113
fobi.contrib.plugins.form_elements.fields.select.apps (module), 108	fobi.contrib.plugins.form_elements.fields.textarea (module), 114
fobi.contrib.plugins.form_elements.fields.select.conf (module), 108	fobi.contrib.plugins.form_elements.fields.textarea.apps (module), 114
fobi.contrib.plugins.form_elements.fields.select.defaults (module), 108	fobi.contrib.plugins.form_elements.fields.time (module), 114
fobi.contrib.plugins.form_elements.fields.select_model_objects (module), 109	fobi.contrib.plugins.form_elements.fields.time.apps (module), 114
fobi.contrib.plugins.form_elements.fields.select_model_objects.conf (module), 109	fobi.contrib.plugins.form_elements.fields.url (module), 115
fobi.contrib.plugins.form_elements.fields.select_model_objects.defaults (module), 109	fobi.contrib.plugins.form_elements.fields.url.apps (module), 114
fobi.contrib.plugins.form_elements.fields.select_model_objects.conf (module), 109	fobi.contrib.plugins.form_elements.security (module), 117
fobi.contrib.plugins.form_elements.fields.select_mptt_model_objects (module), 110	fobi.contrib.plugins.form_elements.security.captcha (module), 115
fobi.contrib.plugins.form_elements.fields.select_mptt_model_objects.conf (module), 110	fobi.contrib.plugins.form_elements.security.captcha.apps (module), 115
fobi.contrib.plugins.form_elements.fields.select_mptt_model_objects.defaults (module), 110	fobi.contrib.plugins.form_elements.security.honeypot (module), 116
fobi.contrib.plugins.form_elements.fields.select_mptt_model_objects.conf (module), 110	fobi.contrib.plugins.form_elements.security.honeypot.apps (module), 115
fobi.contrib.plugins.form_elements.fields.select_multiple (module), 111	fobi.contrib.plugins.form_elements.security.honeypot.conf (module), 116
fobi.contrib.plugins.form_elements.fields.select_multiple.apps (module), 110	fobi.contrib.plugins.form_elements.security.honeypot.defaults (module), 116
fobi.contrib.plugins.form_elements.fields.select_multiple.conf (module), 111	fobi.contrib.plugins.form_elements.security.honeypot.fields (module), 116
fobi.contrib.plugins.form_elements.fields.select_multiple.defaults (module), 111	fobi.contrib.plugins.form_elements.security.honeypot.settings (module), 116
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects (module), 112	fobi.contrib.plugins.form_elements.security.recaptcha (module), 117
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf (module), 111	fobi.contrib.plugins.form_elements.security.recaptcha.apps (module), 117
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults (module), 111	fobi.contrib.plugins.form_elements.test (module), 117
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf (module), 112	fobi.contrib.plugins.form_elements.test.dummy (module), 121
fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.defaults (module), 112	fobi.contrib.plugins.form_handlers (module), 119
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects (module), 113	fobi.contrib.plugins.form_handlers.db_store (module), 118
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.conf (module), 112	fobi.contrib.plugins.form_handlers.db_store.apps (module), 118
fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.defaults (module), 112	fobi.contrib.plugins.form_handlers.db_store.conf (module), 118
fobi.contrib.plugins.form_elements.fields.slug (module), 113	fobi.contrib.plugins.form_handlers.db_store.defaults (module), 118
fobi.contrib.plugins.form_elements.fields.slug.apps (module), 113	fobi.contrib.plugins.form_handlers.db_store.settings (module), 118

- fobi.contrib.plugins.form_handlers.http_repost (module), 119
 - fobi.contrib.plugins.form_handlers.http_repost.apps (module), 119
 - fobi.contrib.plugins.form_handlers.mail (module), 121
 - fobi.contrib.plugins.form_handlers.mail.apps (module), 119
 - fobi.contrib.plugins.form_handlers.mail.conf (module), 119
 - fobi.contrib.plugins.form_handlers.mail.defaults (module), 119
 - fobi.contrib.plugins.form_handlers.mail.fields (module), 120
 - fobi.contrib.plugins.form_handlers.mail.helpers (module), 120
 - fobi.contrib.plugins.form_handlers.mail.settings (module), 121
 - fobi.contrib.plugins.form_handlers.mail.widgets (module), 121
 - fobi.contrib.plugins.form_importers (module), 121
 - fobi.contrib.themes (module), 128
 - fobi.contrib.themes.bootstrap3 (module), 123
 - fobi.contrib.themes.bootstrap3.apps (module), 123
 - fobi.contrib.themes.bootstrap3.widgets (module), 123
 - fobi.contrib.themes.bootstrap3.widgets.form_elements (module), 123
 - fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3 (module), 122
 - fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3.apps (module), 122
 - fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3 (module), 122
 - fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3.apps (module), 122
 - fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3 (module), 123
 - fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3.apps (module), 123
 - fobi.contrib.themes.djangocms_admin_style_theme (module), 124
 - fobi.contrib.themes.djangocms_admin_style_theme.apps (module), 124
 - fobi.contrib.themes.djangocms_admin_style_theme.widgets (module), 124
 - fobi.contrib.themes.djangocms_admin_style_theme.widgets (module), 124
 - fobi.contrib.themes.djangocms_admin_style_theme.widgets (module), 124
 - fobi.contrib.themes.djangocms_admin_style_theme.widgets (module), 124
 - fobi.contrib.themes.foundation5 (module), 127
 - fobi.contrib.themes.foundation5.apps (module), 127
 - fobi.contrib.themes.foundation5.widgets (module), 126
 - fobi.contrib.themes.foundation5.widgets.form_elements (module), 126
 - fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5 (module), 125
 - fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5 (module), 125
 - fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5 (module), 125
 - fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5 (module), 125
 - fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5 (module), 126
 - fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5 (module), 126
 - fobi.contrib.themes.foundation5.widgets.form_handlers (module), 126
 - fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5 (module), 126
 - fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5 (module), 126
 - fobi.contrib.themes.simple (module), 128
 - fobi.contrib.themes.simple.apps (module), 128
 - fobi.contrib.themes.simple.widgets (module), 127
 - fobi.contrib.themes.simple.widgets.form_handlers (module), 127
 - fobi.contrib.themes.simple.widgets.form_handlers.db_store (module), 127
 - fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps (module), 127
 - fobi.data_structures (module), 132
 - fobi.defaults (module), 135
 - fobi.defaults (module), 135
 - fobi.exceptions (module), 135
 - fobi.bootstrap3 (module), 128
 - fobi.management (module), 129
 - fobi.bootstrap3.widgets.apps (module), 128
 - fobi.settings (module), 137
 - fobi.templatetags (module), 131
 - fobi.templatetags.fobi_tags (module), 129
 - fobi.templatetags.future_compat (module), 130
 - fobi.urls (module), 131
 - fobi.validators (module), 137
 - fobi.widgets (module), 137
 - FormCacheError, 136
 - FormElementPluginDoesNotExist, 136
 - FormHandlerPluginError, 136
 - FormHandlerPluginDoesNotExist, 136
 - FormHandlerPluginError, 136
 - FormPluginError, 136
- G**
- get_crop_filter() (in module fobi.contrib.plugins.form_elements.content.content_image.helper

97
get_fobi_form_handler_plugin_custom_actions() (in module fobi.contrib.plugins.form_handlers.db_store.conf), 118
get_fobi_plugin() (in module fobi.contrib.plugins.form_handlers.mail.conf), 119
get_form_field_type() (in module fobi.contrib.plugins.form_handlers.db_store.conf), 118
get_form_hidden_fields_errors() (in module fobi.contrib.plugins.form_handlers.mail.conf), 119
get_setting() (in module fobi.conf), 132
get_setting() (in module fobi.contrib.apps.djangocms_integration.conf), 93
get_setting() (in module fobi.contrib.apps.feincms_integration.conf), 94
get_setting() (in module fobi.contrib.apps.mezzanine_integration.conf), 95
get_setting() (in module fobi.contrib.plugins.form_elements.content.content_image.conf), 97
get_setting() (in module fobi.contrib.plugins.form_elements.content.content_video.conf), 99
get_setting() (in module fobi.contrib.plugins.form_elements.fields.checkbox_multiple.conf), 100
get_setting() (in module fobi.contrib.plugins.form_elements.fields.file.conf), 103
get_setting() (in module fobi.contrib.plugins.form_elements.fields.radio.conf), 107
get_setting() (in module fobi.contrib.plugins.form_elements.fields.select.conf), 108
get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_model_object.conf), 109
get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_multiple.conf), 110
get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf), 111
get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf), 111
get_setting() (in module fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.conf), 112
get_setting() (in module fobi.contrib.plugins.form_elements.security.honeypot.conf), 116
get_setting() (in module fobi.contrib.plugins.form_handlers.db_store.conf), 118
get_setting() (in module fobi.contrib.plugins.form_handlers.mail.conf), 119
H
handle_uploaded_file() (in module fobi.contrib.plugins.form_elements.content.content_image.helper), 97
has_edit_form_entry_permissions() (in module fobi.contrib.plugins.form_handlers.mail.conf), 129
HoneypotField (class in fobi.contrib.plugins.form_elements.security.honeypot.fields), 116
I
ImproperlyConfigured, 135
input_type() (fobi.widgets.NumberInput attribute), 137
insert() (fobi.data_structures.SortableDict method), 132
insert_after_key() (fobi.data_structures.SortableDict method), 132
insert_before_key() (fobi.data_structures.SortableDict method), 133
is_selected() (fobi.widgets.NumberInput attribute), 135
is_hidden() (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget attribute), 121
items() (fobi.data_structures.SortableDict method), 133
iteritems() (fobi.data_structures.SortableDict method), 133
iterkeys() (fobi.data_structures.SortableDict method), 133
itervalues() (fobi.data_structures.SortableDict method), 133
K
keys() (fobi.data_structures.SortableDict method), 133
L
label() (fobi.apps.Config attribute), 132
label() (fobi.contrib.apps.djangocms_integration.apps.Config attribute), 93
label() (fobi.contrib.apps.feincms_integration.apps.Config attribute), 94
label() (fobi.contrib.apps.mezzanine_integration.apps.Config attribute), 95
label() (fobi.contrib.plugins.form_elements.content.content_image.apps.Config attribute), 97
label() (fobi.contrib.plugins.form_elements.content.content_text.apps.Config attribute), 98
label() (fobi.contrib.plugins.form_elements.content.content_video.apps.Config attribute), 98

label (fobi.contrib.plugins.form_elements.fields.boolean.apps.Config attribute), 99

label (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps.Config attribute), 100

label (fobi.contrib.plugins.form_elements.fields.date.apps.Config attribute), 101

label (fobi.contrib.plugins.form_elements.fields.date_drop_down.apps.Config attribute), 101

label (fobi.contrib.plugins.form_elements.fields.datetime.apps.Config attribute), 101

label (fobi.contrib.plugins.form_elements.fields.decimal.apps.Config attribute), 102

label (fobi.contrib.plugins.form_elements.fields.email.apps.Config attribute), 102

label (fobi.contrib.plugins.form_elements.fields.file.apps.Config attribute), 103

label (fobi.contrib.plugins.form_elements.fields.float.apps.Config attribute), 103

label (fobi.contrib.plugins.form_elements.fields.hidden.apps.Config attribute), 104

label (fobi.contrib.plugins.form_elements.fields.input.apps.Config attribute), 105

label (fobi.contrib.plugins.form_elements.fields.integer.apps.Config attribute), 105

label (fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config attribute), 106

label (fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config attribute), 106

label (fobi.contrib.plugins.form_elements.fields.password.apps.Config attribute), 106

label (fobi.contrib.plugins.form_elements.fields.radio.apps.Config attribute), 107

label (fobi.contrib.plugins.form_elements.fields.regex.apps.Config attribute), 108

label (fobi.contrib.plugins.form_elements.fields.select.apps.Config attribute), 108

label (fobi.contrib.plugins.form_elements.fields.select_model_objects.apps.Config attribute), 109

label (fobi.contrib.plugins.form_elements.fields.select_mptt_model_objects.apps.Config attribute), 110

label (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps.Config attribute), 110

label (fobi.contrib.plugins.form_elements.fields.select_multiple_model_objects.apps.Config attribute), 111

label (fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_objects.apps.Config attribute), 112

label (fobi.contrib.plugins.form_elements.fields.slug.apps.Config attribute), 113

label (fobi.contrib.plugins.form_elements.fields.text.apps.Config attribute), 113

label (fobi.contrib.plugins.form_elements.fields.textarea.apps.Config attribute), 114

label (fobi.contrib.plugins.form_elements.fields.time.apps.Config attribute), 114

label (fobi.contrib.plugins.form_elements.fields.url.apps.Config attribute), 114

label (fobi.contrib.plugins.form_elements.security.captcha.apps.Config attribute), 115

label (fobi.contrib.plugins.form_elements.security.honeypot.apps.Config attribute), 115

label (fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config attribute), 117

label (fobi.contrib.plugins.form_elements.test.dummy.apps.Config attribute), 117

label (fobi.contrib.plugins.form_handlers.db_store.apps.Config attribute), 118

label (fobi.contrib.plugins.form_handlers.http_repost.apps.Config attribute), 119

label (fobi.contrib.plugins.form_handlers.mail.apps.Config attribute), 119

label (fobi.contrib.themes.bootstrap3.apps.Config attribute), 123

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3.apps.Config attribute), 122

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3.apps.Config attribute), 122

label (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3.apps.Config attribute), 123

label (fobi.contrib.themes.djangocms_admin_style_theme.apps.Config attribute), 124

label (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.djangocms_admin_style_theme.apps.Config attribute), 124

label (fobi.contrib.themes.foundation5.apps.Config attribute), 127

label (fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5.apps.Config attribute), 125

label (fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5.apps.Config attribute), 125

label (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5.apps.Config attribute), 126

label (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5.apps.Config attribute), 126

label (fobi.contrib.themes.simple.apps.Config attribute), 128

label (fobi.contrib.themes.simple.widgets.form_handlers.db_store.simple.apps.Config attribute), 127

M

media (fobi.widgets.BooleanRadioSelect attribute), 138

media (fobi.widgets.NumberInput attribute), 137

message (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField attribute), 120

move_after_key() (fobi.data_structures.SortableDict method), 133

move_before_key() (fobi.data_structures.SortableDict method), 133

MultiEmailField	(class	in	name (fobi.contrib.plugins.form_elements.fields.regex.apps.Config attribute), 108
	fobi.contrib.plugins.form_handlers.mail.fields),		name (fobi.contrib.plugins.form_elements.fields.select.apps.Config attribute), 108
MultiEmailWidget	(class	in	name (fobi.contrib.plugins.form_elements.fields.select_model_object.apps.Config attribute), 109
	fobi.contrib.plugins.form_handlers.mail.widgets),		name (fobi.contrib.plugins.form_elements.fields.select_mptt_model_object.apps.Config attribute), 110
N			
name (fobi.apps.Config attribute), 132			name (fobi.contrib.plugins.form_elements.fields.select_multiple.apps.Config attribute), 110
name (fobi.contrib.apps.djangocms_integration.apps.Config attribute), 93			name (fobi.contrib.plugins.form_elements.fields.select_multiple_model_object.apps.Config attribute), 111
name (fobi.contrib.apps.feincms_integration.apps.Config attribute), 94			name (fobi.contrib.plugins.form_elements.fields.select_multiple_mptt_model_object.apps.Config attribute), 112
name (fobi.contrib.apps.mezzanine_integration.apps.Config attribute), 95			name (fobi.contrib.plugins.form_elements.fields.slug.apps.Config attribute), 113
name (fobi.contrib.plugins.form_elements.content.content_image.apps.Config attribute), 97			name (fobi.contrib.plugins.form_elements.fields.text.apps.Config attribute), 113
name (fobi.contrib.plugins.form_elements.content.content_text.apps.Config attribute), 98			name (fobi.contrib.plugins.form_elements.fields.textarea.apps.Config attribute), 114
name (fobi.contrib.plugins.form_elements.content.content_video.apps.Config attribute), 98			name (fobi.contrib.plugins.form_elements.fields.time.apps.Config attribute), 114
name (fobi.contrib.plugins.form_elements.fields.boolean.apps.Config attribute), 99			name (fobi.contrib.plugins.form_elements.fields.url.apps.Config attribute), 115
name (fobi.contrib.plugins.form_elements.fields.checkbox_select_multiple.apps.Config attribute), 100			name (fobi.contrib.plugins.form_elements.security.captcha.apps.Config attribute), 115
name (fobi.contrib.plugins.form_elements.fields.date.apps.Config attribute), 101			name (fobi.contrib.plugins.form_elements.security.honeypot.apps.Config attribute), 115
name (fobi.contrib.plugins.form_elements.fields.date_drop_down.apps.Config attribute), 101			name (fobi.contrib.plugins.form_elements.security.recaptcha.apps.Config attribute), 117
name (fobi.contrib.plugins.form_elements.fields.datetime.apps.Config attribute), 101			name (fobi.contrib.plugins.form_elements.test.dummy.apps.Config attribute), 117
name (fobi.contrib.plugins.form_elements.fields.decimal.apps.Config attribute), 102			name (fobi.contrib.plugins.form_handlers.db_store.apps.Config attribute), 118
name (fobi.contrib.plugins.form_elements.fields.email.apps.Config attribute), 102			name (fobi.contrib.plugins.form_handlers.http_repost.apps.Config attribute), 119
name (fobi.contrib.plugins.form_elements.fields.file.apps.Config attribute), 103			name (fobi.contrib.plugins.form_handlers.mail.apps.Config attribute), 119
name (fobi.contrib.plugins.form_elements.fields.float.apps.Config attribute), 103			name (fobi.contrib.themes.bootstrap3.apps.Config attribute), 123
name (fobi.contrib.plugins.form_elements.fields.hidden.apps.Config attribute), 104			name (fobi.contrib.themes.bootstrap3.widgets.form_elements.date_bootstrap3.apps.Config attribute), 122
name (fobi.contrib.plugins.form_elements.fields.input.apps.Config attribute), 105			name (fobi.contrib.themes.bootstrap3.widgets.form_elements.datetime_bootstrap3.apps.Config attribute), 122
name (fobi.contrib.plugins.form_elements.fields.integer.apps.Config attribute), 105			name (fobi.contrib.themes.bootstrap3.widgets.form_elements.dummy_bootstrap3.apps.Config attribute), 123
name (fobi.contrib.plugins.form_elements.fields.ip_address.apps.Config attribute), 106			name (fobi.contrib.themes.djangocms_admin_style_theme.apps.Config attribute), 124
name (fobi.contrib.plugins.form_elements.fields.null_boolean.apps.Config attribute), 106			name (fobi.contrib.themes.djangocms_admin_style_theme.widgets.form_handlers.db_store.apps.Config attribute), 124
name (fobi.contrib.plugins.form_elements.fields.password.apps.Config attribute), 106			name (fobi.contrib.themes.foundation5.apps.Config attribute), 127
name (fobi.contrib.plugins.form_elements.fields.radio.apps.Config attribute), 107			name (fobi.contrib.themes.foundation5.widgets.form_elements.date_foundation5.apps.Config attribute), 125

name (fobi.contrib.themes.foundation5.widgets.form_elements.datetime_foundation5_widget.apps.Config attribute), [125](#)
 name (fobi.contrib.themes.foundation5.widgets.form_elements.dummy_foundation5_widget.apps.Config attribute), [126](#)
 name (fobi.contrib.themes.foundation5.widgets.form_handlers.db_store_foundation5_widget.apps.Config attribute), [126](#)
 name (fobi.contrib.themes.simple.apps.Config attribute), [128](#)
 name (fobi.contrib.themes.simple.widgets.form_handlers.db_store.apps.Config attribute), [127](#)
 NoDefaultThemeSet, [136](#)
 NumberInput (class in fobi.widgets), [137](#)

P

permissions_required() (in module fobi.decorators), [134](#)
 PluginDoesNotExist, [136](#)
 pop() (fobi.data_structures.SortableDict method), [134](#)
 popitem() (fobi.data_structures.SortableDict method), [134](#)
 prep_value() (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget method), [121](#)

R

render() (fobi.contrib.plugins.form_handlers.mail.widgets.MultiEmailWidget method), [121](#)
 render_auth_link() (in module fobi.templatetags.fobi_tags), [129](#)

S

send_mail() (in module fobi.contrib.plugins.form_handlers.mail.helpers), [120](#)
 setdefault() (fobi.data_structures.SortableDict method), [134](#)
 SortableDict (class in fobi.data_structures), [132](#)

T

ThemeDoesNotExist, [136](#)
 to_python() (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField method), [120](#)

U

update() (fobi.data_structures.SortableDict method), [134](#)
 url_exists() (in module fobi.validators), [137](#)

V

validate() (fobi.contrib.plugins.form_handlers.mail.fields.MultiEmailField method), [120](#)
 value_for_index() (fobi.data_structures.SortableDict method), [134](#)
 values() (fobi.data_structures.SortableDict method), [134](#)