
django-fusionbox-contact Documentation

Release 0.0.1

Fusionbox

January 29, 2014

1	Introduction	3
2	Contact	5
2.1	Intro	5
2.2	Installation	5
2.3	Settings	5
2.4	Models	5
2.5	Views	6
2.6	URLS	6
2.7	Templates	6
2.8	Admin	7
2.9	Recipients	7
3	Indices and tables	9
	Python Module Index	11

Contents:

Introduction

Django app for reusable Contact-Us style application. Intended to be a fully functional production level application, but with enough room for customization to suite individual project needs.

Contact the development team at [Fusionbox](#) for questions.

2.1 Intro

The contact module is designed to be a DRY style contact form.

2.2 Installation

- Add `'fusionbox.contact'` to your `INSTALLED_APPS`
- Run `./manage.py migrate` to create the necessary database tables
- Create the required templates (see templates and views below)
- Add urls to your url conf

2.3 Settings

The contact module has the following settings values which which may be used to customize its behavior

CONTACT_FORM_RECIPIENTS Iterable of email addresses. Each person in this list will be emailed for each contact form recipient. If not present, the contact module will use the values present in the Recipients table. (See the Recipients section)

CONTACT_SUBMISSION_VERBOSE_NAME and CONTACT_SUBMISSION_VERBOSE_NAME_PLURAL
Override default `verbose_name` and `verbose_name_plural` meta settings on the `Submission` model.

CONTACT_RECIPIENT_VERBOSE_NAME and CONTACT_RECIPIENT_VERBOSE_NAME_PLURAL Override default `verbose_name` and `verbose_name_plural` meta settings on the `Recipient` model.

2.4 Models

The contact application provides the following models.

```
class fusionbox.contact.models.Recipient (*args, **kwargs)
    Recipient(id, name, email, is_active)
```

```
class fusionbox.contact.models.Submission (*args, **kwargs)
    Submission(id, name, email, comment)
```

2.5 Views

The contact module provides the following class-based views located in `fusionbox.contact.views`

2.6 URLS

You may include the urls for the contact module one of two ways.

1. Include the built in url conf somewhere in your site url conf:

```
url(r'^contact-us/', include('fusionbox.contact.urls')),
```

2. Manually include the urls for both the `index` and `success` views. Often this is the best way to customize the contact form or add extra context variables. The example below is equivalent to the include statement:

```
(r'^contact-us/$', 'fusionbox.contact.views.index', name='contact_index'),  
(r'^contact-us/success/$', 'fusionbox.contact.views.success', name='contact_success'),
```

2.7 Templates

The contact module requires you to create two templates.

2.7.1 Primary Contact Form Template

Default Location: `contact/index.html`

This template is rendered with the context variable `form`.

Example:

```
<form method="post">  
  {% csrf_token %}  
  {{ form }}  
  {% uncaptcha %}  
  <button type="submit">Submit</button>  
</form>
```

2.7.2 Success Page Template

Default Location: `contact/success.html`

Example:

```
<p>Thank you for contacting Us. Someone will be in touch with you shortly!</p>
```

2.7.3 Email Template

Default Location: `mail/contact_form_submission.html`

Successful contact form submissions will be emailed using the `fusionbox.send_markdown_email` function to a list of recipients. The contact module will first look for `CONTACT_FORM_RECIPIENTS` in the settings file, and if not will use the values from the Recipients table.

Default Template:

```

---
subject: Someone has filled out the contact form
---

Someone has submitted the contact form.

- *Name:* {{ submission.name }}
- *Email:* {{ submission.email }}
- *Comment:* {{ submission.comment }}

Use the following link to view this submission.

[{{ host }}{{ submission.get_absolute_url }}]({{ host }}{{ submission.get_absolute_url }})

```

2.8 Admin

The contact app automatically registers the following admin classes for its models.

class `fusionbox.contact.admin.RecipientAdmin` (*model, admin_site*)
 Default admin class for `fusionbox.contact.models.Recipient`. Only registered if no hard-coded list of recipients is found in `settings.CONTACT_FORM_RECIPIENTS`.

class `fusionbox.contact.admin.SubmissionAdmin` (*model, admin_site*)
 Default admin class for `fusionbox.contact.models.Submission`. Allows for bulk csv export of Submission entries, and restricts readonly access.

class `Media`

Ensures that whitespace linebreaks are respected when displaying the `Submission.comment` field.

`SubmissionAdmin.get_readonly_fields` (*request, obj=None*)

Dynamically build a list of field names for the model. This allows for monkeypatches to the `Submission` model to not require a new admin class.

`SubmissionAdmin.has_add_permission` (**args, **kwargs*)

Submissions should not be creatable from the admin center.

`SubmissionAdmin.list_comment` (*submission*)

Allows for display of a truncated version of `Submission.comment` in the `list_display`.

2.9 Recipients

The contact module has two methods for designating recipients to be emailed with the details from contact form submissions. If the `CONTACT_FORM_RECIPIENTS` value is present in the settings file, those recipients will be used.

If the setting is not present, the Recipients model will be registered for the admin site, and the values there will be used.

Indices and tables

- *genindex*
- *modindex*
- *search*

f

fusionbox.contact.admin, 7
fusionbox.contact.models, 5