
Emencia Django Forum Documentation

Release 0.7.3

Emencia

December 21, 2014

1	Features	3
2	Links	5
3	Table of contents	7
3.1	Install	7
3.2	Usage	10
3.3	Changelog	12

It is simple and more suited for an internal team use (like a company's extranet) than for big public communities. By the way all the view are restricted at less to authenticated users, there is no access for anonymous.

Features

- Have **categories** that contains **threads** that contains **messages**;
- Have **thread watches**: users can subscribe to receive email notification for each new message in a thread, see *Thread watch*;
- Have thread **sticky mode** and **announce mode**;
- **i18n** usage for the interface;
- Global or '**per object**' **moderation** on categories, threads and messages, see *Permissions*;
- Optional **markup syntax** for messages, default is *RestructuredText* from docutils but you can use your own, see *Text markup*;
- Optional **DjangoCodeMirror editor** or your own editor if you want, see *Text markup*;

Links

- Read the documentation on [Read the docs](#);
- Download his [PyPi package](#);
- Clone it on his [Github repository](#);

Table of contents

3.1 Install

3.1.1 Requires

- Django `>= 1.5`;
- `autobreadcrumbs` `>= 1.0`;
- `django-braces` `>= 1.2.0,<1.4`;
- `django-crispy-forms` `>= 1.4.0`;
- `django-guardian` `>= 1.2.0`;

Optionnally

- `South` to perform database migrations for next releases;
- If you want to use the shipped *Text markup* integration :
 - `rstview` `>= 0.2`;
 - `Django-CodeMirror` `>= 0.9.7`;
- If you want to display Authors Gravatar in the thread's post list:
 - `django-gravatar2` `>= 1.1.4`;

Procedure

Install it from PyPi:

```
pip install emencia-django-forum
```

Add it to your installed apps in settings:

```
INSTALLED_APPS = (  
    ...  
    'autobreadcrumbs',  
    'guardian',  
    'forum',  
    ...  
)
```

Add its settings (in your project settings):

```
from forum.settings import *
```

(Also you can override some of its settings, see `forum.settings` for more details).

Add `django-guardian` settings (see its doc for more details):

```
AUTHENTICATION_BACKENDS = (
    'django.contrib.auth.backends.ModelBackend', # this is default
    'guardian.backends.ObjectPermissionBackend',
)

ANONYMOUS_USER_ID = None
```

Then register `autobreadcrumbs` context processor in settings:

```
TEMPLATE_CONTEXT_PROCESSORS = (
    ...
    'autobreadcrumbs.context_processors.AutoBreadcrumbsContext',
    ...
)
```

Finally mount its urls and add the `autobreadcrumbs` *autodiscover* in your main `urls.py`:

```
import autobreadcrumbs
autobreadcrumbs.autodiscover()

urlpatterns = patterns('',
    ...
    (r'^forum/', include('forum.urls', namespace='forum')),
    ...
)
```

3.1.2 About autobreadcrumbs

`autobreadcrumbs` automatically build the bread crumbs from the current page path.

If you don't want to use it, you have two choices:

- Simply ignore it, it will be used for automatic page titles but just override `forum`'s base template to remove it;
- If you don't install it, you will have to remove it from your settings and urls, then overrides all `forum`'s template that use its tags;

3.1.3 Text markup

Default behavior configured in settings is to not use any Markup syntax usage.

But if you want you can configure some settings to use a Markup syntax renderer and a form field to use a specific editor.

This can be done with the following settings:

```
# Text markup renderer
FORUM_TEXT_MARKUP_RENDER = None # Default, no renderer

# Field helper for text in forms
FORUM_TEXT_FIELD_HELPER_PATH = None # Default, just a CharField
```

```
# Template to init some Javascript for text in forms
FORUM_TEXT_FIELD_JS_TEMPLATE = None # Default, no JS template

# Validator helper for Post.text in forms
FORUM_TEXT_VALIDATOR_HELPER_PATH = None # Default, no markup validation
```

They are the default values in the forum settings.

Explanations

FORUM_TEXT_FIELD_HELPER_PATH a function that will be used to define a form field to use for text.

Signature is `get_text_field(form_instance, **kwargs)` where:

- `form_instance` is the Form instance where it will be used from;
- `kwargs` is a dict containing all default named arguments to give to the field. These default arguments are `label` for the field label name and `required` that is `True` (you should never change this);

This should return an instantiated form field that must act as a `CharField`.

FORUM_TEXT_VALIDATOR_HELPER_PATH

A function that will be used to clean value on the form field text;

Signature is `clean_restructuredtext(form_instance, content)` where:

- `form_instance` is the Form instance where it will be used from;
- `content` is the value to validate;

Act like a Django form field cleaner method, this should return the cleaned value and eventually raise a validation error if needed.

FORUM_TEXT_MARKUP_RENDER_TEMPLATE

A template to include to render text value with some markup syntax. It will have access to the page context with an additional value named `content` that will be the text to render;

FORUM_TEXT_FIELD_JS_TEMPLATE

A template to include with forms when your custom form field require some Javascript to initialize it. It will have access to page context with an additional value named `field` that will be the targeted form field;

All these settings are only used with forms and template managing `forum.models.Post.text` and `forum.models.Category.description` models attributes.

Example

There are the settings to use the shipped Markup syntax renderer and editor, disabled by default but that you can easily enable in your settings:

```
# Field helper for text in forms
FORUM_TEXT_FIELD_HELPER_PATH = "forum.markup.get_text_field" # Use DjangoCodeMirror

# Validator helper for Post.text in forms
FORUM_TEXT_VALIDATOR_HELPER_PATH = "forum.markup.clean_restructuredtext" # Validation for RST syntax

# Template to init some Javascript for text in forms
FORUM_TEXT_FIELD_JS_TEMPLATE = "forum/markup/_text_field_djangocodemirror_js.html" # Use DjangoCodeMirror
```

```
# Text markup renderer
FORUM_TEXT_MARKUP_RENDER_TEMPLATE = "forum/markup/_text_markup_render.html" # Use Rstview renderer
```

Read their source code to see how they work in detail.

Warning: Before enabling these settings you must install `rstview` and `Django-CodeMirror`, see optional requirements to have the right versions to install.

3.1.4 Author informations

In thread's post list, the default behavior for each post is to display the author's username but if you want you can display what you want using an included template.

This app ships a template to enabled you to display the `Gravatar` from the author's email and its username below.

To use it, just add this setting to your settings file:

```
# Template to display author infos in thread's post list
FORUM_AUTHOR_VCARD_TEMPLATE = "forum/author/_vcard.html" # Use Gravatar
```

Warning: Before enabling these settings you must install `django-gravatar2`, see optional requirements to have the right versions to install.

3.2 Usage

3.2.1 Permissions

The forum make usage of `django-guardian` to manage 'per object permissions' or 'global permissions'.

Actually you need to use the Django admin and be a staff user with the right permissions for managing forum objects to add these permissions for your users.

And so, you can add the needed permissions globally to the whole forum within each user accounts. Or you can add a permission for a specific object its edit page using the link named *Object permissions*.

- Users with `forum.add_category` permissions can create categories;
- All users can create a new thread;
- All users can add a new message to a thread that is visible and not closed;
- Users with `forum.moderate_category` can edit them and can manage (edit, delete) their threads parameters and messages. They are called *Category moderators*;
- Users with `forum.moderate_thread` can edit thread parameters, edit thread messages and delete thread messages. They are called *Thread moderators*;
- Users have permission to edit their own message if `settings.FORUM_OWNER_MESSAGE_CAN_EDIT` is `True`;

Others Category's and Thread's model permissions have no roles on frontend.

Permission error response

Permission error is rendered though a `403.html` template that is already embedded within this app, you can override it in your project with adding your custom `403.html` template in your project templates directory.

Also you can use another template name, you will have to define its name in `settings.GUARDIAN_TEMPLATE_403` (this is a setting from [django-guardian](#), see its documentation for more details).

3.2.2 Thread watch

Users can subscribe to watch for new messages on a thread and so they can receive notifications about them.

When a new message is posted on a thread, all users that have subscribed to the thread watch will receive an email, excepting for the message author himself.

`settings.FORUM_EMAIL_SENDER` will be used to send emails if defined, else `settings.DEFAULT_FROM_EMAIL` will be used instead.

You can change the email subject and content templates used to build the emails:

- `forum/threadwatch/email_subject.txt` for the subject;
- `forum/threadwatch/email_content.txt` for the content;

These templates receive a context with some variables:

- `SITE` : the current Site (from the Django “sites” framework);
- `thread_instance` : the thread instance where the message has been posted;
- `post_instance` : the message instance that have been posted;

Create your own email sender for notifications

This is working with [Django](#) signals, when a new thread message is created, a signal is send and a receiver is listen to them. The receiver will receive a signal containing some arguments about the message and the thread watches so it can be used to send email notifications.

The signals usage in this process enable you to make your own receiver to send notifications with your specific email provider/sender or even on another message system (irc, jabber, whatever..).

Default behavior is to use `forum.signals.new_message_posted_receiver` that use simple Django email sending and generally it should fit to your needs.

However if you need to have your own receiver, just define the Python path to it, remember that it should be a callable respecting the defined `kwargs` and avoid to import Forum models in your code as it will make a circular import error.

An example in your settings to use your own receiver:

```
FORUM_NEW_POST_SIGNAL = 'myproject.signals.mycallback'
```

And a receiver example:

```
def new_message_posted_receiver(sender, **kwargs):
    message = kwargs['post_instance']
    threadwatches = kwargs['threadwatches']

    print "New message #{0} has been posted on thread:".format(message.id), message.thread
```

```
for item in threadwatchs:
    print "*", item, "for", item.owner
```

See `forum.signals.new_message_posted_receiver` to have a real example and don't forget to read about signals in the Django documentation.

3.3 Changelog

3.3.1 Version 0.7.3 - 2014/12/21

- Fix MANIFEST file for statics;

3.3.2 Version 0.7.2 - 2014/12/21

- Refactoring templates organization;
- Embed default SCSS sources files and its *compiled* CSS;
- Embed used webfont;

3.3.3 Version 0.7.1 - 2014/11/23

- Add an optional setting and template to display author's Gravatar in thread's post list using `django-gravatar2>=1.1.4`;
- Refactoring how we use the crispies within forms;
- Hide some fields for non moderator in thread forms;

3.3.4 Version 0.7 - 2014/11/23

- Remove **rstview** from required dependancy;
- Changing dependancy `django-crispy-forms >= 1.4.0` to `crispy-forms-foundation>=0.3.6` because we directly this one;
- Changing forms, views and templates to use optional *Text markup*;
- Add shipped stuff (not enabled by default) to use **Django-CodeMirror editor** and **RestructuredText** with the *Text markup* system;
- Moving documentation from README to the `docs/` directory, improve it and publish it on [Read the docs](#);
- Update French translation catalog for minor fix;

3.3.5 Version 0.6.8 - 2014/11/11

- Removes South dependancy;
- Minor improvements on `forum.utils.ListAppendView` class view;

3.3.6 Version 0.6.7 - 2014/08/30

- Move form layouts from `forum.forms.layouts` to `forum.forms.crispies`;
- Fix some bad choices with Permission mixins views;
- Add django-guardian dependancy;

3.3.7 Version 0.6.6 - 2014/08/20

Add catalog translation for French, this close issue #7.

3.3.8 Version 0.6.5 - 2014/08/20

Add a view to find and redirect to the exact page where is a Post to have an absolute url working with paginated list, this close issue #5.

3.3.9 Version 0.6.1 - 2014/08/19

Validate RST syntax on Post message and Category description, this close issue #8.

3.3.10 Version 0.6 - 2014/08/19

- Add 'Django signals' usage when new message is posted so we can use a signal receiver to send email notification;
- Finalize threadwatch with a working email sending and update README for full explanation on threadwatch this close issue #2;

3.3.11 Version 0.5 - 2014/08/17

- Use rstview template filter to render message text into RST, this close issue #3;
- Return 403 response with a rendered template, this close issue #1;
- Update README;

3.3.12 Version 0.4 - 2014/08/16

- Improve README;
- Add the right permission usage with django-guardian;
- Add category and thread moderators;
- Add form confirm into message delete form view;

3.3.13 Version 0.3.1 - 2014/08/12

- Update package dependancies for missing South entry;
- Update README;

3.3.14 Version 0.3 - 2014/08/12

Update to autobreadcrumbs 1.0 to have the full url's namespace support, use namespaces everywhere

3.3.15 Version 0.2 - 2014/08/11

- Use translation strings for everything;
- Finish templates with Foundation;
- Redo some models to have better modified dates;
- Add initial South migrations;
- Add default settings with pagination;
- Add crispy layouts for all forms;
- Some other minor changes;

3.3.16 Version 0.1 - 2014/08/04

First commit with a working version but not fully integrated.