django-roughpages Documentation

Release 0.1.0

Alisue

June 05, 2014

Contents

Author Alisue <lambdalisue@hashnote.net>

Supported python versions Python 2.6, 2.7, 3.2, 3.3

Supported django versions Django 1.2 - 1.6

An template based the flatpages like app. Not like django's flatpages app, django-roughpages render a template file which determined from the accessed URL. It is quite combinient when you want to render simple static page. You do not need to prepare `urls.py` or `views.py` anymore for that kind of simple static page.

django-roughpages call `roughpages.views.roughpage` view with the accessed URL when django raise `Http404` exception. The view automatically find the corresponding template file from `roughpages` directory in one of your template directories. Assume if the user accessed http://localhost/foo/bar/hoge/. If there is no urls pattern patched with the URL, django-roughpages try to find corresponding template file such as `templates/roughpages/foo/bar/hoge.html`. If django-roughpages find the corresponding template file, it will render the template and return the `HttpResponse`, otherwise it re-raise `Http404` exception.

You can complicatedly select the corresponding template file. django-roughpages determine the filename with a backend system. The default backend is `roughpages.backends.AuthTemplateFilenameBackend` and it prefer `hoge_anonymous.html` or `hoge_authenticated.html` than `hoge.html` depends on the accessed user authentication state. Thus you can simply prepare the page for authenticated user as `<something>_authenticated.html` and for anonymous user as `<something>_anonymous.html`.

You can control the backend behavior with making a custom backend. To make a custom backend, you need to inherit `roughpages.backends.TemplateFilenameBackendBase` and override `prepare_filenames(self, filename, request)` method. The method receive an original filename and `HttpRequest` instance and must return a filename list. The django-roughpages then try to load template file from the beginning of the list, thus the order of the appearance is the matter.

CHAPTER 1

Documentations

1.1 Tutorials

django-roughpages have several distincitive features to help you to make static pages. You should follow at least the “Quick tutorial” section to get the image of django-roughpages.

1.1.1 Quick tutorial

If you have not installed django-roughpages yet, you can install it via pip (Mac OS X / Linux) or easy_install (Windows) with:

```
> pip install django-roughpages
> easy_install django-roughpages
```

Ok. Now make a tutorial project with the following command. If you don’t have `django-admin.py`, make sure that you have a latest django in your system:

```
> django-admin.py startproject roughpagestut
```

The command above will create the following files and directories:

```
roughpagestut/
-- manage.py
-- roughpagestut
     -- __init__.py
     -- settings.py
     -- urls.py
     -- wsgi.py
```

Then open `roughpagestut/roughpagestut/settings.py` and edit the settings. Refer the sample settings below (`# ...` indicate the cut out).

```
# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
import os
BASE_DIR = os.path.dirname(os.path.dirname(__file__))

INSTALLED_APPS = (
    # ... lot more
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'roughpages',
)
```

```
# ...
MIDDLEWARE_CLASSES = (
    # ... lot more
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'roughpages.middleware.RoughpageFallbackMiddleware',
)

TEMPLATE_DIRS = (
    # roughpages/templates
    os.path.join(BASE_DIR, 'templates'),
)
```

Ok now, run syncdb and server with the following commands:

```
> python manage.py syncdb
> python manage.py runserver 8000
```

When you access http://localhost:8000/, you will see "It worked!" page. Confirm that accessing http://localhost:8000/info/ lead you to 404 Not found.

Ofcourse, we have not make any views yet thus there are no other pages exists. Now let's make a static page by django-roughpages. Create `info.html` file in `templates/roughpages` directory. The final directory tree would be same as the below:

```
roughpagestut/
-- manage.py
-- roughpagestut
|   -- __init__.py
|   -- settings.py
|   -- urls.py
|   -- wsgi.py
-- templates
    -- roughpages
        -- info.html
```

Then open `info.html` and edit the file as followed

```
<html>
<body>
    <p>Hello Rough!</p>
</body>
</html>
```

Now if you access the http://localhost:8000/info/, you will see "Hello Rough!" page. So django-roughpages automatically detect the corresponding template files (in this case, `info.html`) from the accessed URL. If there is a corresponding template files in the template directory, django-roughpages simply render the template and respond it. Otherwise it re-raise the `Http404` exceptions for other middlewares such as django's flatpages app.

1.1.2 Complex template file finding

django-roughpages use a backend to determine the filename from the accessed url; the default backend is `roughpages.backends.AuthTemplateFilenameBackend` and the backend return two template filenames depends on the accessed users authentication state. Assume that the user accessed http://localhost:8000/info/, if the accessed user is authenticated then the backend return `roughpages/info_authenticated.html` and `roughpages/info.html`. Then django-roughpages try to find the template files from the beginning of the list, thus the order of the appearance is important.

You can find detail informations about built-in backends at *Backends*.

1.2 Configurations

django-roughpages provide the following options.

ROUGHPAGES_BACKEND A class object or dots separated python import path to specify the template filename backend. The default value is `'roughpages.backends.AuthTemplateFilenameBackend'`

ROUGHPAGES_INDEX_FILENAME A filename without extension which is used to represent the root URL (`'/'`). When the django-roughpage process the root URL, this filename is used instead to find the template file like `'/.html'`. The default value is `'index'`

ROUGHPAGES_TEMPLATE_DIR A directory name which is automatically prepended to the template file path; if the user access to http://localhost/foo/bar/hoge, the actual template file path would be `'roughpages/foo/bar/hoge.html'` when the `ROUGHPAGES_TEMPLATE_DIR` is set to `'roughpages'`. The default value is `'roughpages'`

ROUGHPAGES_TEMPLATE_FILE_EXT A file extension which is used to create template file path. The default value is `'.html'`

ROUGHPAGES_RAISE_TEMPLATE_DOES_NOT_EXISTS If this is `True` then `TemplateDoesNotExists` exception would be throwed when a user accessed to invalid url and django-roughpages could not find any corresponding template files. This feature does not work in product mode (`DEBUG=False`) while django-roughpages simply return back the response in exceptions when the `DEBUG` is `False`. The default value is `False`

1.3 Backends

django-roughpages provde the following backends. All backends are located in `roughpages.backends` module.

TemplateFilenameBackendBase A backend interface class. Developer can create a custom backend by inheriting this class. The subclass must override `prepare_filenames(self, normalized_url, request)` method to provide the template filename list.

PlainTemplateFilenameBackend A simple backend. This backend just return the corresponding filename of URL (http://localhost:8000/foo/bar/hoge/ to `'foo/bar/hoge.html'`). However, it will return `['index.html']` when the user accessed to the root of the site (the filename is determined by `settings.ROUGHPAGES_INDEX_FILENAME` value).

AuthTemplateFilenameBackend A backend which lookup the accessed users authentication state. If the user is authenticated, the backend return `<something>_authenticated.html` and `<something>.html`. Otherwise it return `<something>_anonymous.html` and `<something>.html`. Additionally it will return `['index_authenticated.html', 'index.html']` or `['index_anonymous.html', 'index.html']` when the user accessed to the root of the site (the filename is determined by `settings.ROUGHPAGES_INDEX_FILENAME` value).

1.4 src

1.4.1 roughpages package

Subpackages

roughpages.backends package

Submodules

roughpages.backends.auth module

roughpages.backends.base module

roughpages.backends.decorators module

roughpages.backends.plain module

Module contents

roughpages.tests package

Submodules

roughpages.tests.compat module

roughpages.tests.test_backends module

roughpages.tests.test_middleware module

roughpages.tests.test_utils module

roughpages.tests.test_views module

Module contents

Submodules

roughpages.compat module

roughpages.conf module

roughpages.middleware module

roughpages.models module

roughpages.utils module

`roughpages.utils.`**`remove_pardir_symbols`**(*path*, *sep='/'*, *pardir='..'*)

Remove relative path symobls such as '..'

Parameters

- **path** (*str*) – A target path string

- **sep** (*str*) – A strint to refer path delimiter (Default: *os.sep*)
- **pardir** (*str*) – A string to refer parent directory (Default: *os.pardir*)

Returns str

`roughpages.utils.`**`url_to_filename`**(*url*)

Safely translate url to relative filename

Parameters **url** (*str*) – A target url string

Returns str

roughpages.views module

Module contents

django-roughpages

1.5 Indices and tables

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

Python Module Index