# django-calendarium Documentation

*Release 0.2*

**Martin Brochhaus**

**Mar 30, 2017**

# Contents

Contents:

# Installation

If you want to install the latest stable release from PyPi:

```
$ pip install django-calendarium
```

If you feel adventurous and want to install the latest commit from GitHub:

```
$ pip install -e git+git://github.com/bitmazk/django-calendarium.git#egg=calendarium
```

Add `calendarium` (and the `filer` dependencies) to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    ...,
    'filer',
    'mptt',
    'easy_thumbnails',
    'calendarium',
)
```

Add `django.template.context_processors.request` to your `TEMPLATE_CONTEXT_PROCESSORS`:

```
TEMPLATE_CONTEXT_PROCESSORS = (
    ...,
    'django.template.context_processors.request',
)
```

Add the urls to your main `urls.py`:

```
urlpatterns = patterns('',
    ...
    url(r'^calendar/', include('calendarium.urls')),
)
```

If you are using a Django version below 1.7, add the following setting:

**SOUTH_MIGRATION_MODULES = {** 'calendarium': 'calendarium.south_migrations',

}

Run the migrations:

```
./manage.py migrate
```

# Usage

First think about the layout you want to integrate. In most cases you might want to add the calendar for the current month first. Reverse with:

```
{% url "calendar_current_month" %}
```

Then use the different URLs to let the user navigate through months, weeks and days. The event management should be intuitive. Be sure to add `Rule` models, if you want to use occurences.

## Template Tags

When using the calendarium template tags in your template, include:

```
{% load calendarium_tags %}
```

We provide a template tag to render a defined amount of upcoming occurrences:

```
{% render_upcoming_events %}
```

The default amount is `5`. You can add your own:

```
{% render_upcoming_events 1000 %}
```

# Contribute

If you want to contribute to this project, please perform the following steps:

```
# Fork this repository
# Clone your fork
$ mkvirtualenv django-calendarium
$ pip install -r requirements.txt
$ pip install -r test_requirements.txt
$ ./runtests.py
# You should get no failing tests

$ git checkout -b feature_branch master
# Implement your feature and tests
# Describe your change in the CHANGELOG.txt
$ git add . && git commit
$ git push -u origin HEAD
# Send us a pull request for your feature branch
```

Whenever you run the tests a coverage output will be generated in `calendarium/tests/coverage/index.html`. When adding new features, please make sure that you keep the coverage at 100%.

## Updating the documentation

In order to update the documentation, just edit the `.rst` files in `docs/source`. If you would like to see your changes in the browser, run `make html` in `docs`.

Check out the reStructuredText Primer for more information on the concepts and syntax.

# CHAPTER 4

## Indices and tables

- genindex
- modindex
- search