

---

# **django-chameleon Documentation**

***Release 0.1-beta***

**Xabier (slok) Larrakoetxea**

**Sep 27, 2017**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
1.1	Dependencies . . . . .	3
1.2	Options . . . . .	3
1.3	Template change methods . . . . .	5
1.4	Use/installation . . . . .	6
<b>2</b>	<b>Util functions</b>	<b>9</b>
2.1	Cookie/session utils . . . . .	9
2.2	GET/POST utils . . . . .	9
2.3	Context utils . . . . .	9
2.4	Path utils . . . . .	9
2.5	Template utils . . . . .	9
<b>3</b>	<b>Changelog</b>	<b>11</b>
3.1	0.1-beta . . . . .	11



**Description:**

This app will let you change the template/theme dinamically in your [Django](#) website. This way the user can select the best theme that fits his likes, browser, resolution... in the website

**Features:**

- Session theme store (the theme sets until the cookie/session is deleted)
- No limit of themes
- Project example

**License:**

django-chameleon is open-source/libre software under the BSD license. You can fork or read the code at: <http://github.com/slok/django-chameleon> and report bugs at: <http://github.com/slok/django-chameleon/issues> .

**Contents:**



### Dependencies

- **Django (Depends on the method)**
  - Loader method: Django  $\geq$  1.2
  - TemplateResponse method: Django = 1.3
- Python  $\geq$  2.4 (I use 2.6)
- Session middleware (database. if you use db session store)
- sphinx [Optional for the docs]

### Options

There are some options to put them in settings.py, a few of them are necessary, the rest are optional.

#### CHAMELEON\_AUTOMATED

- Optional
- Type: Boolean
- Default: True

This option will automate the process of theme changing. If is activated, the theme will be changed in the session variable, in the request context and in the loader or middleware. If the flag is False then the user is the one that has to change the theme, for example in the views, The session and the request context are automaticm, but the loader or the middleware(theme setter) not. Example:

```
CHAMELEON_AUTOMATED = True
```

## CHAMELEON\_FORM\_KEY

- Optional
- Type: String
- Default: 'theme'

This option is to set the name where the theme selection will be placed in the GET and POST Example:

```
CHAMELEON_FORM_KEY = ''
```

## CHAMELEON\_DEFAULT\_THEME

- Optional
- Type: String
- Default: 'default'

This option is to set the default name of the primary or default theme Example:

```
CHAMELEON_DEFAULT_THEME = ''
```

## CHAMELEON\_CONTEXT\_KEY

- Optional
- Type: String
- Default: 'theme'

This option is to set the request context variable name, to access from the templates, for example with `{{ theme }}` Example:

```
CHAMELEON_CONTEXT_KEY = ''
```

## CHAMELEON\_COOKIE\_KEY

- Optional
- Type: String
- Default: 'theme'

This option is to set the session/cookie variable name. Example:

```
CHAMELEON_COOKIE_KEY = ''
```

## DEFAULT\_LEVEL\_CUT

- Optional
- Type: Integer
- Default: 0



If you want to put the default theme in a subfolder where the other themes aren't placed you could say how many levels to remove from the default path (like in patch/diff ;))

Example with no cut level:

```
template root folder: /templates
default theme: /templates/default/*.html
blue theme: /templates/default/blue/*.html
```

Example with cut level 1:

```
template root folder: /templates
default theme: /templates/default/*.html
blue theme: /templates/blue/*.html
```

Example:

```
DEFAULT_LEVEL_CUT = 0
```

## CHAMELEON\_SITE\_THEMES

- **Not optional**
- **Type: Dictionary**
  - key(name): String
  - value(path): String

This option is necessary to know where and wich are the themes. If some theme path is void (‘’) then the app assumes that the path is the name of the theme. The structure inside the theme (f.e: blue) has to be the same as the default one. The path is from the template root folder (normally is templates/)

Example:

```
CHAMELEON_SITE_THEMES = {
    'green': 'themes/green',
    'black': 'themes/black',
    'blue': 'blue',
    'red': '',
}
```

## Template change methods

Chameleon manages two methods to change the themes, `Loader` and the new Django `TemplateResponse` both have good and bad things.

The loader method is the old one, this is the common one that have been used when we wanted to interact with the response of the view before rendering. The other method is new in Django 1.3, This method has been made to interact with the view return data before creating the response and rendering (middleware). The preferred way is the new `templateResponse` method. The differences are this:

### Loader

- No need to change anything(only add the loader in `settings.py`)

- Django 1.2
- Heavier method (in terms of load and time)

## TemplateResponse

- If the views dont return a simpleTemplateResponse or TemplateResponse we have to change
- Django 1.3
- Middleware, so is clean and very lightweight (fast)
- This method is created for this such of things (it was necessary a method like this and finally we can use it)

## Use/installation

### Installation

```
pip install git+git://github.com/slok/django-chameleon.git
```

### Settings variables

First we have to add the *options* that we want in settings.py

---

**Note:** CHAMELEON\_SITE\_THEMES is **not** optional

---

Also we need to add our contextProcessors to TEMPLATE\_CONTEXT\_PROCESSOR this way we can use `{{ theme }}` in templates, to do this:

```
from django.conf.global_settings import TEMPLATE_CONTEXT_PROCESSORS

TEMPLATE_CONTEXT_PROCESSORS += ('chameleon.context_processors.theme',)
```

### Add chameleon to installed apps

In settings add chameleon to INSTALLED\_APPS, for example:

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'chameleon',
)
```

## Add middleware

Add both middlewares to `MIDDLEWARE_CLASSES` (the second one is only for the `TemplateResponse` method, but if loader is activated in `TEMPLATE_LOADERS` the `SetResponseTemplate` middleware disables automatically). Like this:

```
MIDDLEWARE_CLASSES = (  
    'django.middleware.common.CommonMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'chameleon.middleware.DetectTheme',  
    'chameleon.middleware.SetResponseTemplate',  
)
```

## Set method (Loader or TemplateResponse)

We have to select the *Template change methods* to use when Django has to change the themes

---

**Note:** `TemplateResponse` is recommended

---

### Loader

Add our loader to settings (the first one) for example:

```
TEMPLATE_LOADERS = (  
    'chameleon.loader.Loader',  
    'django.template.loaders.filesystem.Loader',  
    'django.template.loaders.app_directories.Loader',  
    # 'django.template.loaders.eggs.Loader',  
)
```

---

**Note:** Place the loader the first one

---

### TemplateResponse

- We don't have to add the loader to `TEMPLATE_LOADERS`
- All the views have to call the template with `TemplateResponse` (or `SimpleTemplateResponse`)



## CHAPTER 2

---

Util functions

---

**Cookie/session utils**

**GET/POST utils**

**Context utils**

**Path utils**

**Template utils**



## CHAPTER 3

---

### Changelog

---

#### 0.1-beta

This isnt a release, is in beta so now is in development, it will be tagged when reaches 0.1 this will be when some tests and documentation are finished