

---

# **django-eximagination Documentation**

*Release 0.8*

**Artur Barseghyan <artur.barseghyan@gmail.com>**

January 29, 2017



<b>1</b>	<b>Prerequisites</b>	<b>3</b>
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage example (in a Django template)</b>	<b>7</b>
<b>4</b>	<b>License</b>	<b>9</b>
<b>5</b>	<b>Support</b>	<b>11</b>
<b>6</b>	<b>Author</b>	<b>13</b>
<b>7</b>	<b>Documentation</b>	<b>15</b>
7.1	eximagination package . . . . .	15
7.1.1	Subpackages . . . . .	15
	eximagination.templatetags package . . . . .	15
	Submodules . . . . .	15
	eximagination.templatetags.eximagnate module . . . . .	15
	Module contents . . . . .	16
7.1.2	Submodules . . . . .	16
7.1.3	eximagination.apps module . . . . .	16
7.1.4	eximagination.conf module . . . . .	16
7.1.5	eximagination.defaults module . . . . .	16
7.1.6	eximagination.helpers module . . . . .	16
7.1.7	eximagination.settings module . . . . .	17
7.1.8	eximagination.tests module . . . . .	17
7.1.9	eximagination.utils module . . . . .	17
7.1.10	Module contents . . . . .	17
<b>8</b>	<b>Indices and tables</b>	<b>19</b>
	<b>Python Module Index</b>	<b>21</b>



Fetch external images directly from templates.

A Django template tag library which allows to download external images, store them locally and return the local path to locally stored image to a desired context variable, along with `width` and `height` attributes of the image fetched. Caches the fetched images locally for the given time (set in settings).

You could, for example, use this app to solve the problems with displaying of a mixed content (assets loaded from HTTP and HTTPS sources).



---

## Prerequisites

---

- Django 1.8, 1.9, 1.10
- Python  $\geq 2.7$ ,  $\geq 3.4$





---

## Installation

---

### 1. Install django-eximagination

Latest stable version on PyPI:

```
pip install django-eximagination
```

Latest stable version from GitHub:

```
pip install https://github.com/barseghyanartur/django-eximagination/archive/stable.tar.gz
```

### 2. Add eximagination to INSTALLED\_APPS.

```
INSTALLED_APPS = (  
    # ...  
    'eximagination',  
    # ...  
)
```

### 3. Configure

By default, django-eximagination expects your files to be stored in `/media/external_images` directory. If location varies, redefine the directories in your Django settings, make sure the path is writable and that `www-data` (or whatever is applicable) has rights to write into it.

```
import os  
  
BASE_DIR = os.path.dirname(os.path.dirname(__file__))  
MEDIA_ROOT = os.path.join(BASE_DIR, '..', '..', 'media')  
EXIMAGINATION_MEDIA_ROOT = os.path.join(MEDIA_ROOT, 'external_images')  
EXIMAGINATION_MEDIA_URL = '/media/external_images/'  
EXIMAGINATION_MEDIA_RELATIVE_ROOT = 'external_images/'  
# After 30 days we re-fetch the file anyway.  
EXIMAGINATION_EXPIRATION_INTERVAL = 2592000
```



---

## Usage example (in a Django template)

---

See the [example directory](#) for working code example.

Example #1:

```
{% load eximagnate %}

```

Example #2:

```
{% load eximagnate thumbnail %}
{% eximagnate 'http://www.google.com/intl/en/images/logo.gif' as original %}

```

In both cases there are two additional context variables added:

- `ei_width` - Width of the image
- `ei_height` - Height of the image



---

**License**

---

GPL 2.0/LGPL 2.1



---

**Support**

---

For any issues contact me at the e-mail given in the *Author* section or open an issue on BitBucket/GitHub.





---

**Author**

---

Artur Barseghyan <[artur.barseghyan@gmail.com](mailto:artur.barseghyan@gmail.com)>



---

## Documentation

---

Contents:

### Table of Contents

- *django-eximagination*
  - *Prerequisites*
  - *Installation*
  - *Usage example (in a Django template)*
  - *License*
  - *Support*
  - *Author*
  - *Documentation*
  - *Indices and tables*

## 7.1 eximagination package

### 7.1.1 Subpackages

**eximagination.templatetags package**

**Submodules**

**eximagination.templatetags.eximagnate module**

`eximagination.templatetags.eximagnate.eximagnate` (*parser, token*)

Copy an external image locally.

To just output the absolute url to the thumbnail:

```
{% load eximagnate %}


```

To put the the downloaded image URL on the context instead of just rendering the relative url, finish the tag with `as [context_var_name]`:

```
{% load eximagnate thumbnail %}
```

```
{% eximagine 'http://www.google.com/intl/en/images/logo.gif' as original %}

```

Eximagination will automatically add “ei\_width” and “ei\_height” variables to the content - those are the original width and height values of the obtained image.

TODO: Implement force\_update feature (some extra tokens parsing)

## Module contents

### 7.1.2 Submodules

#### 7.1.3 eximagination.apps module

Apps.

```
class eximagination.apps.Config(app_name, app_module)
    Bases: django.apps.config.AppConfig
    Config.
    label = 'eximagination'
    name = 'eximagination'
```

#### 7.1.4 eximagination.conf module

```
eximagination.conf.get_setting(setting, override=None)
    Get setting.
```

Get a setting from eximagination conf module, falling back to the default.

If override is not None, it will be used instead of the setting.

##### Parameters

- **setting** – String with setting name
- **override** – Value to use when no setting is available. Defaults to None.

**Returns** Setting value.

#### 7.1.5 eximagination.defaults module

#### 7.1.6 eximagination.helpers module

```
eximagination.helpers.project_dir(base)
    Get path to directory/file from current path.
```

```
eximagination.helpers.PROJECT_DIR(base)
    Get path to directory/file from current path.
```

## 7.1.7 eximagination.settings module

## 7.1.8 eximagination.tests module

**class** `eximagination.tests.EximaginationUtilsTest` (*methodName='runTest'*)

Bases: `unittest.case.TestCase`

Tests of `eximagination.utils.obtain_image` function.

**setUp** ()

Set up.

**test\_01\_obtain\_image** (\*args, \*\*kwargs)

`eximagination.tests.log_info` (*func*)

Log some useful info.

## 7.1.9 eximagination.utils module

`eximagination.utils.obtain_image` (*image\_source='', save\_to='', media\_url='', force\_update=False, expiration\_interval=None, debug=False*)

Get an image from absolute url and saves it locally.

Check whether image already exists in local directory and only if not - try to download it and save it. Validate validity of the image (rely on PIL Image class validation).

### Parameters

- **image\_source** (*str*) –
- **save\_to** (*str*) –
- **media\_url** (*str*) –
- **force\_update** (*bool*) –
- **expiration\_interval** (*int*) – Expiration interval in seconds.
- **debug** (*bool*) –

**Return list** (*relative\_url\_of\_the\_image, original\_image\_width, original\_image\_height*)

## 7.1.10 Module contents

A Django template tag library which allows to download external images, store them locally and return the local path to locally stored image to a desired context variable, along with *width* and *height* of the image fetched. Caches the fetched images locally for the given time (set in settings). You could, for example, use this app to solve the problems with displaying of a mixed content (assets loaded from HTTP and HTTPS sources).

- **default\_app\_config**: Default Django app config.

`eximagination.obtain_image` (*image\_source='', save\_to='', media\_url='', force\_update=False, expiration\_interval=None, debug=False*)

Get an image from absolute url and saves it locally.

Check whether image already exists in local directory and only if not - try to download it and save it. Validate validity of the image (rely on PIL Image class validation).

### Parameters

- **image\_source** (*str*) –
- **save\_to** (*str*) –
- **media\_url** (*str*) –
- **force\_update** (*bool*) –
- **expiration\_interval** (*int*) – Expiration interval in seconds.
- **debug** (*bool*) –

**Return list** (relative\_url\_of\_the\_image, original\_image\_width, original\_image\_height)

---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`





**e**

eximagination, [17](#)  
eximagination.apps, [16](#)  
eximagination.conf, [16](#)  
eximagination.defaults, [16](#)  
eximagination.helpers, [16](#)  
eximagination.settings, [17](#)  
eximagination.templatetags, [16](#)  
eximagination.templatetags.eximagnate,  
[15](#)  
eximagination.tests, [17](#)  
eximagination.utils, [17](#)



**C**

Config (class in `eximagination.apps`), 16

**E**

`eximagnate()` (in module `eximagination.templatetags.eximagnate`), 15

`eximagination` (module), 17

`eximagination.apps` (module), 16

`eximagination.conf` (module), 16

`eximagination.defaults` (module), 16

`eximagination.helpers` (module), 16

`eximagination.settings` (module), 17

`eximagination.templatetags` (module), 16

`eximagination.templatetags.eximagnate` (module), 15

`eximagination.tests` (module), 17

`eximagination.utils` (module), 17

`EximaginationUtilsTest` (class in `eximagination.tests`), 17

**G**

`get_setting()` (in module `eximagination.conf`), 16

**L**

`label` (`eximagination.apps.Config` attribute), 16

`log_info()` (in module `eximagination.tests`), 17

**N**

`name` (`eximagination.apps.Config` attribute), 16

**O**

`obtain_image()` (in module `eximagination`), 17

`obtain_image()` (in module `eximagination.utils`), 17

**P**

`PROJECT_DIR()` (in module `eximagination.helpers`), 16

`project_dir()` (in module `eximagination.helpers`), 16

**S**

`setUp()` (`eximagination.tests.EximaginationUtilsTest` method), 17

**T**

`test_01_obtain_image()` (`eximagination.tests.EximaginationUtilsTest` method), 17