
django-dummy-thumbnails

Documentation

Release 0.2.3

Artur Barseghyan <artur.barseghyan@gmail.com>

Jun 04, 2019

Contents

1	Prerequisites	3
2	Installation	5
3	Usage	7
3.1	Replace broken images with dummy ones	7
3.1.1	Supported thumbnailers	7
3.1.1.1	django-imagekit	7
3.1.1.2	sorl-thumbnail	8
3.1.1.3	easy-thumbnails	9
4	Dealing with broken or invalid dummy images	11
5	Importing images from feed	13
5.1	Installation	13
5.2	Usage	13
5.3	Configuration	14
6	Demo	15
6.1	Run demo locally	15
7	Testing	17
8	License	19
9	Support	21
10	Author	23
11	Documentation	25
11.1	dummy_thumbnails package	26
11.1.1	Subpackages	26
11.1.1.1	dummy_thumbnails.contrib package	26
11.1.1.1.1	Subpackages	26
11.1.1.1.1.1	dummy_thumbnails.contrib.image_importers package	26
11.1.1.1.1.2	Subpackages	26
11.1.1.1.1.3	dummy_thumbnails.contrib.image_importers.feed package	26
11.1.1.1.1.4	Subpackages	26

11.1.1.1.1.5	dummy_thumbnails.contrib.image_importers.feed.management package	26
11.1.1.1.1.6	Subpackages	26
11.1.1.1.1.7	dummy_thumbnails.contrib.image_importers.feed.management.commands package	26
11.1.1.1.1.8	Submodules	26
11.1.1.1.1.9	dummy_thumbnails.contrib.image_importers.feed.management.commands.dummy_thumbnails module	26
11.1.1.1.1.10	Module contents	26
11.1.1.1.1.11	Module contents	26
11.1.1.1.1.12	Submodules	26
11.1.1.1.1.13	dummy_thumbnails.contrib.image_importers.feed.apps module . .	26
11.1.1.1.1.14	dummy_thumbnails.contrib.image_importers.feed.conf module . .	27
11.1.1.1.1.15	dummy_thumbnails.contrib.image_importers.feed.defaults module	27
11.1.1.1.1.16	dummy_thumbnails.contrib.image_importers.feed.reader module .	27
11.1.1.1.1.17	dummy_thumbnails.contrib.image_importers.feed.settings module	27
11.1.1.1.1.18	Module contents	27
11.1.1.1.1.19	Module contents	29
11.1.1.1.1.20	dummy_thumbnails.contrib.thumbnailers package	29
11.1.1.1.1.21	Subpackages	29
11.1.1.1.1.22	dummy_thumbnails.contrib.thumbnailers.django_imagekit package	29
11.1.1.1.1.23	Submodules	29
11.1.1.1.1.24	dummy_thumbnails.contrib.thumbnailers.django_imagekit.generatorlibrary module	29
11.1.1.1.1.25	Module contents	29
11.1.1.1.1.26	dummy_thumbnails.contrib.thumbnailers.easy_thumbnails package	29
11.1.1.1.1.27	Submodules	29
11.1.1.1.1.28	dummy_thumbnails.contrib.thumbnailers.easy_thumbnails.source_generators module	29
11.1.1.1.1.29	Module contents	29
11.1.1.1.1.30	dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail package	29
11.1.1.1.1.31	Subpackages	29
11.1.1.1.1.32	dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail.engines package	29
11.1.1.1.1.33	Submodules	29
11.1.1.1.1.34	dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail.engines.dummy_engine module	29
11.1.1.1.1.35	Module contents	29
11.1.1.1.1.36	Module contents	29
11.1.1.1.1.37	Module contents	29
11.1.1.1.2	Module contents	29
11.1.1.2	dummy_thumbnails.management package	29
11.1.1.2.1	Subpackages	29
11.1.1.2.1.1	dummy_thumbnails.management.commands package	29
11.1.1.2.1.2	Submodules	29
11.1.1.2.1.3	dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images module	29
11.1.1.2.1.4	dummy_thumbnails.management.commands.dummy_thumbnails_verify_dummy_images module	30
11.1.1.2.1.5	Module contents	30
11.1.1.2.2	Module contents	30
11.1.1.3	dummy_thumbnails.tests package	30
11.1.1.3.1	Submodules	30
11.1.1.3.2	dummy_thumbnails.tests.base module	30

11.1.1.3.3	dummy_thumbnails.tests.helpers module	30
11.1.1.3.4	dummy_thumbnails.tests.test_core module	31
11.1.1.3.5	Module contents	31
11.1.2	Submodules	31
11.1.3	dummy_thumbnails.apps module	31
11.1.4	dummy_thumbnails.base module	31
11.1.5	dummy_thumbnails.conf module	31
11.1.6	dummy_thumbnails.defaults module	31
11.1.7	dummy_thumbnails.helpers module	32
11.1.8	dummy_thumbnails.settings module	32
11.1.9	Module contents	32
12	Indices and tables	33
	Python Module Index	35
	Index	37

Dummy thumbnails for *most popular* Django thumbnail generators.

There are times when you have a database of a Django site and you need to quickly start it up to fix/develop, but then you realise that images are missing and you need to have images, because either your layout is broken or, in the worst case, entire site is broken (500). This library has been written for developers in order to avoid above mentioned problems in the shortest way possible, with least efforts possible.

CHAPTER 1

Prerequisites

- Django 1.8, 1.9, 1.10, 1.11, 2.0
- Python 2.7, 3.4, 3.5, 3.6

Although `django-dummy-thumbnails` is not being tested against older versions of Django, tests do pass with Django versions 1.5, 1.6 and 1.7.

CHAPTER 2

Installation

- (1) Install in your virtual environment

Latest stable version from PyPI:

```
pip install django-dummy-thumbnails
```

or latest stable version from GitHub:

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

or latest stable version from BitBucket:

```
pip install https://bitbucket.org/barseghyanartur/django-dummy-thumbnails/get/  
↪stable.tar.gz
```

- (2) Add `dummy_thumbnails` to your `INSTALLED_APPS` in the global `settings.py`.

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

- (3) Collect the statics:

```
./manage.py collectstatic --noinput
```

- (4) Specify a custom path to your images directory in settings:

```
DUMMY_THUMBNAILS_IMAGES_PATH = '/home/path/to/images/'
```

This should be a directory inside your media directory of your Django project. Otherwise Django would raise a `SuspiciousOperation` exception. In order not to duplicate tons of files for each project, you are advised to create symlinks to the images directory in the media directory of your Django project.

```
ln -s /home/path/to/images /home/repos/your-django-project/media
```

If you prefer to use included [public domain images](#), run the following management command:

```
./manage.py dummy_thumbnails_symlink_dummy_images
```

And specify the following path in settings:

```
DUMMY_THUMBNAILS_IMAGES_PATH = os.path.join(MEDIA_ROOT, 'mixed')
```

Common usage examples.

3.1 Replace broken images with dummy ones

That's what it's all about - replacing the broken images with dummy ones.

Note: You should **never** use this in production. All the changes mentioned above and below are supposed to be applied to **development** settings only.

3.1.1 Supported thumbnailers

Most popular image thumbnailers for Django ([django-imagekit](#), [sorl-thumbnail](#) and [easy-thumbnails](#)) are supported. If you can't find your favourite thumbnailer, open an issue or consider making a pull request.

3.1.1.1 django-imagekit

Integration with [django-imagekit](#).

Modify your settings in the following way:

- (1) Add `imagekit`, `dummy_thumbnails` and `dummy_thumbnails.contrib.thumbnailers.django_imagekit.generatorlibrary` to the `INSTALLED_APPS`:

```
INSTALLED_APPS = [  
    # ...  
    'imagekit',  
    'dummy_thumbnails',  
    'dummy_thumbnails.contrib.thumbnailers.django_imagekit.generatorlibrary',  
]
```

(continues on next page)

(continued from previous page)

```
# ...  
]
```

- (2) Collect the statics:

```
./manage.py collectstatic --noinput
```

- (3) Specify a custom path to your images directory in settings:

```
DUMMY_THUMBNAILS_IMAGES_PATH = os.path.join(MEDIA_ROOT, 'mixed')
```

- (4) If you are using the included public domain images, don't forget to create a symlink:

```
./manage.py dummy_thumbnails_symlink_dummy_images
```

- (5) Now the following would work:

```
{% load imagekit %}  
  
{% thumbnail '640x480' 'None1' %}  
{% thumbnail '480x640' 'None2' %}  
{% thumbnail '200x200' 'None3' %}
```

3.1.1.2 sorl-thumbnail

Integration with [sorl-thumbnail](#).

Modify your settings in the following way:

- (1) Add `sorl.thumbnail` and `dummy_thumbnails` to the `INSTALLED_APPS`:

```
INSTALLED_APPS = [  
    # ...  
    'sorl.thumbnail',  
    'dummy_thumbnails',  
    # ...  
]
```

- (2) Set the dummy thumbnail engine as `THUMBNAIL_ENGINE`:

```
THUMBNAIL_ENGINE = 'dummy_thumbnails.contrib.sorl_thumbnail.engines.  
↳DummyThumbnailsEngine'
```

- (3) Collect the statics:

```
./manage.py collectstatic --noinput
```

- (4) Specify a custom path to your images directory in settings:

```
DUMMY_THUMBNAILS_IMAGES_PATH = os.path.join(MEDIA_ROOT, 'mixed')
```

- (5) If you are using the included public domain images, don't forget create a symlink:

```
./manage.py dummy_thumbnails_symlink_dummy_images
```

- (6) Now the following would work:

```
{% load thumbnail %}

{% thumbnail 'None1' "640x480" crop="center" as im %}
    
{% endthumbnail %}

{% thumbnail 'None2' "480x640" crop="center" as im %}
    
{% endthumbnail %}

{% thumbnail 'None3' "200x200" crop="center" as im %}
    
{% endthumbnail %}
```

3.1.1.3 easy-thumbnails

Integration with [easy-thumbnails](#).

Modify your settings in the following way:

- (1) Add `easy_thumbnails` and `dummy_thumbnails` to the `INSTALLED_APPS`:

```
INSTALLED_APPS = [
    # ...
    'easy_thumbnails',
    'dummy_thumbnails',
    # ...
]
```

- (2) Add dummy thumbnail generator to `THUMBNAIL_SOURCE_GENERATORS`:

```
THUMBNAIL_SOURCE_GENERATORS = (
    'dummy_thumbnails.contrib.thumbnailers.easy_thumbnails.source_generators.'
    '↔dummy_thumbnail',
)
```

- (3) Collect the statics:

```
./manage.py collectstatic --noinput
```

- (4) Specify a custom path to your images directory in settings:

```
DUMMY_THUMBNAILS_IMAGES_PATH = os.path.join(MEDIA_ROOT, 'mixed')
```

- (5) If you are using the included public domain images, don't forget to create a symlink:

```
./manage.py dummy_thumbnails_symlink_dummy_images
```

- (6) Now the following would work:

```
{% load thumbnail %}




```

Dealing with broken or invalid dummy images

Of course, it's always better to have a good working set of dummy images. However, it might happen that for some reason one of your dummy images is broken.

The recommended approach is to use a management command `dummy_thumbnails_verify_dummy_images`, which has been written in order to verify the dummy images and identify possible problems. It also lets you remove broken/invalid dummy images.

To remove broken/invalid dummy images with confirmation, type:

```
./manage.py dummy_thumbnails_verify_dummy_images
```

To remove broken/invalid dummy images without confirmation, type:

```
./manage.py dummy_thumbnails_verify_dummy_images --noinput
```

To just list broken/invalid dummy images without removal, type:

```
./manage.py dummy_thumbnails_verify_dummy_images --list
```

Another way to avoid failures is to set the value of `DUMMY_THUMBNAILS_VERIFY_IMAGES` to `True` in your project settings. Beware, that this slows down the start up time of your Django project, although does not slow down further rendering of the images.

Importing images from feed

Imports images from feeds, that support enclosures.

5.1 Installation

Add `dummy_thumbnails.contrib.image_importers.feed` to your `INSTALLED_APPS` in the global `settings.py`.

```
INSTALLED_APPS = (  
    # ...  
    'dummy_thumbnails.contrib.image_importers.feed',  
    # ...  
)
```

5.2 Usage

To import 50 dummy images from Flickr [commercialphotos](#) feed, type:

```
./manage.py dummy_thumbnails_import_from_feed
```

You can provide a custom feed URL and the number of dummy images to import. To import 100 images from Flickr [nationalgeographic](#) group, type:

```
./manage.py dummy_thumbnails_import_from_feed \  
    "https://api.flickr.com/services/feeds/groups_pool.gne?id=36256495@N00" \  
    --limit=100
```

5.3 Configuration

As you have seen, syntax allows to read images from any feed (that supports enclosures). In your project, you might want to make it easy for developers, so that they don't have to type the feed URL. Therefore a setting `DUMMY_THUMBNAILS_FEED_DEFAULT_FEED_URL` has been introduced. It defaults to the URL of the [commercialphotos](#) group of the Flickr.

```
DUMMY_THUMBNAILS_FEED_DEFAULT_FEED_URL = "https://api.flickr.com/" \
                                           "services/feeds/groups_pool.gne" \
                                           "?id=36256495@N00"
```

6.1 Run demo locally

In order to be able to quickly evaluate the *django-dummy-thumbnails*, a demo app (with a quick installer) has been created (works on Ubuntu/Debian, may work on other Linux systems as well, although not guaranteed). Follow the instructions below to have the demo running within a minute.

Grab the latest `dummy_thumbnails_demo_installer.sh`:

```
wget -O - https://raw.githubusercontent.com/barseghyanartur/django-dummy-thumbnails/stable/  
→examples/dummy_thumbnails_demo_installer.sh | bash
```

Open your browser and test the app.

- URL: <http://127.0.0.1:8001/>

If quick installer doesn't work for you, see the manual steps on running the [example project](#).

CHAPTER 7

Testing

Simply type:

```
./runtests.py
```

or use tox:

```
tox
```

or use tox to check specific env:

```
tox -e py35
```

or run Django tests:

```
./manage.py test dummy_thumbnails --settings=settings.testing
```


CHAPTER 8

License

GPL 2.0/LGPL 2.1

CHAPTER 9

Support

For any issues contact me at the e-mail given in the *Author* section.

CHAPTER 10

Author

Artur Barseghyan <artur.barseghyan@gmail.com>

Contents:

Table of Contents

- *django-dummy-thumbnails*
 - *Prerequisites*
 - *Installation*
 - *Usage*
 - * *Replace broken images with dummy ones*
 - *Supported thumbnailers*
 - *django-imagekit*
 - *sorl-thumbnail*
 - *easy-thumbnails*
 - *Dealing with broken or invalid dummy images*
 - *Importing images from feed*
 - * *Installation*
 - * *Usage*
 - * *Configuration*
 - *Demo*
 - * *Run demo locally*
 - *Testing*
 - *License*

- *Support*
- *Author*
- *Documentation*
- *Indices and tables*

11.1 dummy_thumbnails package

11.1.1 Subpackages

11.1.1.1 dummy_thumbnails.contrib package

11.1.1.1.1 Subpackages

11.1.1.1.1.1 dummy_thumbnails.contrib.image_importers package

11.1.1.1.1.2 Subpackages

11.1.1.1.1.3 dummy_thumbnails.contrib.image_importers.feed package

11.1.1.1.1.4 Subpackages

11.1.1.1.1.5 dummy_thumbnails.contrib.image_importers.feed.management package

11.1.1.1.1.6 Subpackages

11.1.1.1.1.7 dummy_thumbnails.contrib.image_importers.feed.management.commands package

11.1.1.1.1.8 Submodules

11.1.1.1.1.9 dummy_thumbnails.contrib.image_importers.feed.management.commands.dummy_thumbnails_imp module

11.1.1.1.1.10 Module contents

11.1.1.1.1.11 Module contents

11.1.1.1.1.12 Submodules

11.1.1.1.1.13 dummy_thumbnails.contrib.image_importers.feed.apps module

Apps.

```
class dummy_thumbnails.contrib.image_importers.feed.apps.Config(app_name,  
                                                             app_module)
```

Bases: `django.apps.config.AppConfig`

Config.

```
label = 'dummy_thumbnails_contrib_image_importers_feed'
name = 'dummy_thumbnails.contrib.image_importers.feed'
```

11.1.1.1.14 dummy_thumbnails.contrib.image_importers.feed.conf module

Conf module.

- `get_setting`: Get app setting.

```
dummy_thumbnails.contrib.image_importers.feed.conf.get_setting(setting, override=None)
```

Get setting.

Get a setting from `dummy_thumbnails` 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.

11.1.1.1.15 dummy_thumbnails.contrib.image_importers.feed.defaults module

Defaults module.

- `DEFAULT_FEED_URL`: URL of the default feed.

11.1.1.1.16 dummy_thumbnails.contrib.image_importers.feed.reader module

11.1.1.1.17 dummy_thumbnails.contrib.image_importers.feed.settings module

Settings module.

- `DEFAULT_FEED_URL`: URL of the default feed.

11.1.1.1.18 Module contents

Feed image importer for `django-dummy-thumbnails`. Imports photos from feeds, that support enclosures.

- `default_app_config`: Default Django app config.

11.1.1.1.1.19 Module contents

11.1.1.1.1.20 `dummy_thumbnails.contrib.thumbnailers` package

11.1.1.1.1.21 Subpackages

11.1.1.1.1.22 `dummy_thumbnails.contrib.thumbnailers.django_imagekit` package

11.1.1.1.1.23 Submodules

11.1.1.1.1.24 `dummy_thumbnails.contrib.thumbnailers.django_imagekit.generatorlibrary` module

11.1.1.1.1.25 Module contents

11.1.1.1.1.26 `dummy_thumbnails.contrib.thumbnailers.easy_thumbnails` package

11.1.1.1.1.27 Submodules

11.1.1.1.1.28 `dummy_thumbnails.contrib.thumbnailers.easy_thumbnails.source_generators` module

11.1.1.1.1.29 Module contents

11.1.1.1.1.30 `dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail` package

11.1.1.1.1.31 Subpackages

11.1.1.1.1.32 `dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail.engines` package

11.1.1.1.1.33 Submodules

11.1.1.1.1.34 `dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail.engines.dummy_engine` module

11.1.1.1.1.35 Module contents

11.1.1.1.1.36 Module contents

11.1.1.1.1.37 Module contents

11.1.1.1.2 Module contents

11.1.1.2 `dummy_thumbnails.management` package

11.1.1.2.1 Subpackages

11.1.1.2.1.1 `dummy_thumbnails.management.commands` package

11.1.1.2.1.2 Submodules

11.1.1.2.1.2.1 `dummy_thumbnails` package

11.1.1.2.1.2.1.1 `dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images` module

- Command: Management command. Prepare the dirs and create symlinks for testing purposes.

```
class dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images.Command(st  
st  
no  
  
    Bases: django.core.management.base.BaseCommand  
    Prepare dirs and symlinks.  
    handle (*args, **options)  
        Handle.  
    help = 'Prepare dirs and make symlinks to dummy images'
```

11.1.1.2.1.4 dummy_thumbnails.management.commands.dummy_thumbnails_verify_dummy_images module

11.1.1.2.1.5 Module contents

11.1.1.2.2 Module contents

11.1.1.3 dummy_thumbnails.tests package

11.1.1.3.1 Submodules

11.1.1.3.2 dummy_thumbnails.tests.base module

Base tests module.

- log_info: Method decorator, logs info about the method (name, return value).
- LOG_INFO: Flag.
- app_setup: Setup the app.
- skip: Method decorator. If used, tests is skipped.
- is_app_setup_completed: Check if app setup has been completed.
- mark_app_setup_as_completed: Mark the app setup as completed.

```
dummy_thumbnails.tests.base.log_info(func)  
    Log some useful info.
```

```
dummy_thumbnails.tests.base.skip(func)  
    Simply skip the test.
```

```
dummy_thumbnails.tests.base.is_app_setup_completed()  
    Check if app setup is completed.
```

```
dummy_thumbnails.tests.base.mark_app_setup_as_completed()  
    Mark app setup as completed.
```

11.1.1.3.3 dummy_thumbnails.tests.helpers module

Base tests module.

- setup_app: Setup the app, collect the static, mark app setup as completed.

`dummy_thumbnails.tests.helpers.setup_app` (*collectstatic=False*)
Set up it8next.

11.1.1.3.4 `dummy_thumbnails.tests.test_core` module

11.1.1.3.5 Module contents

11.1.2 Submodules

11.1.3 `dummy_thumbnails.apps` module

Apps.

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

11.1.4 `dummy_thumbnails.base` module

11.1.5 `dummy_thumbnails.conf` module

Conf module.

- `get_setting`: Get app setting.

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

Get a setting from `dummy_thumbnails.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.

11.1.6 `dummy_thumbnails.defaults` module

Defaults module.

- `IMAGES_PATH`: Path to random images.
- `VERIFY_IMAGES`: If set to `True`, images are verified.

11.1.7 dummy_thumbnails.helpers module

Helpers module.

- `prepare_dirs_and_symlinks`: Prepare dirs and symlinks.

`dummy_thumbnails.helpers.prepare_dirs_and_symlinks` (*create_dirs=False*)

Prepare dirs and symlinks.

Parameters `create_dirs` (*bool*) – If set to True, dirs are created.

11.1.8 dummy_thumbnails.settings module

Settings module.

- `IMAGES_PATH`: Path to random images.
- `VERIFY_IMAGES`: If set to True, images are verified.

11.1.9 Module contents

Dummy thumbnails for most popular Django thumbnail generators.

- `default_app_config`: Default Django app config.

CHAPTER 12

Indices and tables

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

d

- `dummy_thumbnails`, 32
- `dummy_thumbnails.apps`, 31
- `dummy_thumbnails.conf`, 31
- `dummy_thumbnails.contrib`, 29
 - `image_importers`, 29
 - `feed`, 27
 - `apps`, 26
 - `conf`, 27
 - `defaults`, 27
 - `management`, 26
 - `commands`, 26
 - `settings`, 27
 - `thumbnailers`, 29
 - `django_imagekit`, 29
 - `easy_thumbnails`, 29
 - `sorl_thumbnail`, 29
 - `engines`, 29
- `defaults`, 31
- `helpers`, 32
- `management`, 30
 - `commands`, 30
 - `dummy_thumbnails_symlink_dummy_images`, 29
- `settings`, 32
- `dummy_thumbnails.tests`, 31
 - `base`, 30
 - `helpers`, 30

C

Command (*class in dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images*), 30

Config (*class in dummy_thumbnails.apps*), 31

Config (*class in dummy_thumbnails.contrib.image_importers.feed.apps*), 26

D

dummy_thumbnails (*module*), 32

dummy_thumbnails.apps (*module*), 31

dummy_thumbnails.conf (*module*), 31

dummy_thumbnails.contrib (*module*), 29

dummy_thumbnails.contrib.image_importers (*module*), 29

dummy_thumbnails.contrib.image_importers.feed (*module*), 27

dummy_thumbnails.contrib.image_importers.feed.apps (*module*), 26

dummy_thumbnails.contrib.image_importers.feed.conf (*module*), 27

dummy_thumbnails.contrib.image_importers.feed.defaults (*module*), 27

dummy_thumbnails.contrib.image_importers.feed.management (*module*), 26

dummy_thumbnails.contrib.image_importers.feed.management.commands (*module*), 26

dummy_thumbnails.contrib.image_importers.feed.settings_completed() (*in module dummy_thumbnails.tests.base*), 30

dummy_thumbnails.contrib.thumbnailers (*module*), 29

dummy_thumbnails.contrib.thumbnailers.django_imagekit (*module*), 29

dummy_thumbnails.contrib.thumbnailers.easy_thumbnail (*module*), 29

dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail (*module*), 29

dummy_thumbnails.contrib.thumbnailers.sorl_thumbnail.engines (*module*), 29

dummy_thumbnails.defaults (*module*), 31

dummy_thumbnails.helpers (*module*), 32

dummy_thumbnails.management (*module*), 30

dummy_thumbnails.management.commands.dummy_thumbnails (*module*), 30

dummy_thumbnails.settings (*module*), 32

dummy_thumbnails.tests (*module*), 31

dummy_thumbnails.tests.base (*module*), 30

dummy_thumbnails.tests.helpers (*module*), 30

G

get_setting() (*in module dummy_thumbnails.conf*), 31

get_setting() (*in module dummy_thumbnails.contrib.image_importers.feed.conf*), 27

handle() (*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

H

handle() (*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

help(*dummy_thumbnails.management.commands.dummy_thumbnails_symlink_dummy_images* attribute), 30

N

`name` (*dummy_thumbnails.apps.Config* attribute), [31](#)

`name` (*dummy_thumbnails.contrib.image_importers.feed.apps.Config* attribute), [27](#)

P

`prepare_dirs_and_symlinks()` (in module *dummy_thumbnails.helpers*), [32](#)

S

`setup_app()` (in module *dummy_thumbnails.tests.helpers*), [30](#)

`skip()` (in module *dummy_thumbnails.tests.base*), [30](#)