django-ultimatethumb

unknown

Feb 10, 2022

CONTENTS

1 Requirements **3**

2 Prepare for development **5**

3 Resources **7**

3.1 Installation 7

3.2 Usage 8

3.3 Changelog 9

3.4 API Reference 11

4 Indices and tables **15**

Python Module Index **17**

Index **19**

django-ultimatethumb is another Django library for generating thumbnails but has some advantages:

- Thumbnails are not generated when the templatetag is called. Instead, images are generated on demand when they are requested by the browser. This can lead to a major speedup of your page response times.
- Thumbnails can be generated from static files too (for example to downscale retina-optimized images and therefore reducing traffic).
- Generate multiple thumbnail sizes at once for use in *picture* html tags with multiple sources (e.g. with media queries).

CHAPTER

ONE

REQUIREMENTS

django-ultimatethumb supports Python 3 only and requires at least Django 1.11.

CHAPTER

TWO

PREPARE FOR DEVELOPMENT

A Python 3.6 interpreter is required in addition to pipenv.

```
$ poetry install
```

Now you're ready to run the tests:

```
$ make tests
```

CHAPTER

THREE

RESOURCES

- Documentation
- Bug Tracker
- Code

Contents:

3.1 Installation

To install *django-ultimatethumb* just use your preferred Python package installer:

```
$ pip install django-ultimatethumb
```

Add some stuff to your Django settings

```
INSTALLED_APPS = (
    # some other apps
    'ultimatethumb',
)

# This is the path where the generated thumbnail files are cached.
ULTIMATETHUMB_ROOT = '/filesystem/path/to/thumbnails/'
# This is the base url for your thumbnails
ULTIMATETHUMB_URL = '/thumbnails/'
```

Next, add the *django-ultimatethumb* urls to your `urls.py`

```
urlpatterns += patterns(
    '',
    url(
        r'^{0}/'.format(settings.ULTIMATETHUMB_URL.strip('/')),
        include('ultimatethumb.urls')
    ),
```

Hint: You can use the `ULTIMATETHUMB_URL` setting in your `urls.py` to make sure that the urls are in sync.

3.2 Usage

To use *django-ultimatethumb* in your templates, just load the templatetags and call the `ultimatethumb` tag with proper parameters:

```
{% load ultimatethumb_tags %}
{% ultimatethumb 'mythumb' mymodel.imagefield.name sizes='200x0,400x0' %}
<picture>
{% for source in mythumb %}
    <source
        srcset="{{ source.url_2x }} 2x, {{ source.url }} 1x"
        {% if not forloop.last %}media="(max-width: {{ source.viewport.width }}px)"{%␣
↪endif %}
    />
    {% if forloop.last %}<img src="{{ source.url }}" />{% endif %}
{% endfor %}
</picture>
```

This gives you a full-featured picture tag including multiple sources with media queries for different browser sizes and also provides retina images.

You can also use *django-ultimatethumb* in a much simpler way:

```
{% load ultimatethumb_tags %}
{% ultimatethumb 'mythumb' mymodel.imagefield.name sizes='400x0' %}
<img src="{{ mythumb.0.url }}" />
```

To resize static images, just prefix the path with `static:`, for example:

```
{% load ultimatethumb_tags %}
{% ultimatethumb 'mythumb' 'static:img/logo.jpg' sizes='400x0' %}
<img src="{{ mythumb.0.url }}" />
```

There are many other options/parameters to pass to the templatetag. Please refer to the codebase until the documentation is more complete.

You can also pass the viewport size in addition to the requested thumbnail size:

```
{% load ultimatethumb_tags %}
{% ultimatethumb 'mythumb' 'static:img/logo.jpg' sizes='400x0:600x0' %}
<img src="{{ mythumb.0.url }}" />
```

This will set the *thumbnail.viewport.width* to 600.

If you want so save some characters, you might short cut the sizes by leaving out the "x0" for the auto'ed dimesion.

```
{% load ultimatethumb_tags %}
{% ultimatethumb 'mythumb' 'static:img/logo.jpg' sizes='400:600' %}
<img src="{{ mythumb.0.url }}" />
```

The sizes are now the same as if you would use sizes='400x0,600x0'.

3.2.1 Options

You can pass some options to the thumbnail tag:

- upscale: Configures if the input should be upscaled if requested sizes are larger than source.
- retina: Option to enable retina support (by providing both url and url_2x)
- crop: Deside if images should be cropped if requested sizes doesn't fit source aspect ratio.
- quality: Configures quality for image compression
- pngquant: Configures the pngquant compression factor

Hint: The *crop* option can be set to True for default gravity when cropping (which is *Center*). You can also pass valid GraphicsMagick gravities (North, NorthEeast, East, SouthEast, ...) or their abbreviation (N, NE, E, SE, ...)

3.3 Changelog

3.3.1 1.2.0 - 2021-09-28

- Added support for Django 3.0, 3.1, 3.2
- Dropped support for Django < 2.2
- Added Black formatting
- Moved to poetry

3.3.2 1.1.0 - 2019-03-19

- Improve performance when using static files as source (use stored_name instead of hashed_name to get the path)

3.3.3 1.0.0 - 2018-08-08

- Removed barbeque dependency, now used python-command-executor
- Dropped support for Django < 1.11
- Dropped support for Python 2
- Moved to pipenv based development

3.3.4 0.8.0 - 2017-12-05

- Add support for base64 output of thumbnails

3.3.5 0.7.0 - 2017-11-03

- Fixed thumbnail size when cropping with exact target sizes
- Added support for Django 1.11

3.3.6 0.6.0 - 2017-07-31

- Added viewport support to pass viewport max-width/max-height in addition to size
- Added support for crop gravity ("crop" now accepts True (fallback to "Center") or a gravity orientation according to the GraphicsMagick documentation)

3.3.7 0.5.0 - 2017-02-14

- Fix bug when using static: sources in DEBUG=False and not using django-compressor
- Drop support for Django <1.8

3.3.8 0.4.1 - 2016-05-04

- Added support for Django 1.9

3.3.9 0.3.0 - 2015-11-03

- Added pngquant support
- Bugfix for retina / factor2x mode
- Improved option passing in templatetag.

3.3.10 0.2.0 - 2015-10-20

- Added domain support to allow serving of thumbnails from a different domain
- Fixed handling of staticfiles when using CachedStaticFilesStorage
- Bugfix for path quoting
- Added Django 1.8 support

3.3.11 0.1.0 - 2015-03-24

- Initial release.

Api documentation:

3.4 API Reference

3.4.1 Storage module

class `ultimatethumb.storage.`**`ThumbnailFileSystemStorage`**(**args*, ***kwargs*)

Bases: `django.core.files.storage.FileSystemStorage`

Extended FileSystemStorage from Django which uses ULTIMATETHUMB_* settings to initialize storage backend.

This storage is used to handle the generated files.

class `ultimatethumb.storage.`**`ThumbnailStorage`**

Bases: `django.utils.functional.LazyObject`

Lazy class to defer the initialization of the thumbnail_storage to give settings a chance to override the used storage backend.

3.4.2 Templatetags module

`ultimatethumb.templatetags.ultimatethumb_tags.`**`ultimatethumb`**(*context*, *as_var*, *source*, *sizes=None*, *upscale=False*, *crop=None*, *retina=True*, *quality=None*, *pngquant=None*)

Main template tag to generate thumbnail sourcesets.

3.4.3 Thumbnail module

class `ultimatethumb.thumbnail.`**`Size`**(*width*, *height*)

Bases: `tuple`

`height`

Alias for field number 1

`width`

Alias for field number 0

class `ultimatethumb.thumbnail.`**`ThumbnailSet`**(*source*, *sizes*, *options*)

Bases: `object`

A ThumbnailSet holds the source configuration and a number of thumbnails as requested.

`__init__`(*source*, *sizes*, *options*)

Takes a valid source and a list of requested sizes together with additional options.

`thumbnails`

The thumbnails property returns a list of available thumbnails.

`get_source_size`()

Returns the file size of the source.

If retina option is enabled, pretend that the source is half as large as it is. We do this to ensure that we have "retina" images which effectively are doubled in size. Doing this, we never have to upscale the image.

`get_sizes`()

Parse the given sizes and return a list of size requests.

get_thumbnails()
Returns a list of thumbnails based on the requested sizes. Some additional calculations are done to make sure we don't return thumbnails larger than the source file is.

class ultimatethumb.thumbnail.**Thumbnail**(*source*, *opts*)
Bases: object

This object represents a single thumbnail.

__init__(*source*, *opts*)
Given a source and options, this method initializes the Thumbnail object. Some validation on the provided options are done.

classmethod from_name(*name*)
Using a name, reconstruct a thumbnail object. The name is actually a cache key which is used by get_thumb_data to fetch the original thumbnail configuration.

get_name()
Generate the thumbnail name for the current thumbnail configuration.

size
The size property returns the calculated, estimated thumbnail size without actually generating the thumbnail image.

viewport
Returns the valid viewport for this thumbnail, might be used in templates to configure the source sets properly.

property url
The url property is responsible for returning the acutal thumbnail url.

property url_2x
Returns the retina url for the thumbnail if retina is enabled.

property base64
Returns the base64 representation of the thumbnail to use in a src attribute.

property requested_size
Returns the requested thumbnail size.

exists(*factor=1*)
Checks if the thumbnail already exists.

get_mimetype()
Returns the mime type of the thumbnail based on the thumbnail file name.

get_storage_url(*factor=1*)
Returns the real url to use in the ultimatethumb view for returning the image.

get_storage_path(*factor=1*, *generate=True*)
Returns the storage path in filesystem of the thumbnail file.

get_size(*factor=1*)
Returns the actual generated thumbnail image size.

get_estimated_size()
Calculates the estimated thumbnail image dimensions based on the source size and the options provided.

get_storage_name(*factor=1*, *suffix=None*)
Returns the name to use when storing the thumbnail to disk.

generate(*factor=1*)
Genrate the thumbnail using Graphicsmagick and Pngquant (if enabled and source image is a png file.

get_gm_options(*factor=1*)
Generates the option set dor Graphicsmagick to generate the thumbnail.

get_base64_content()
Reads the base64 version of the thumbnail from disk and returns content.

get_base64_path(*generate=True*)
Calculates the path of the base64 image version. If path does not exist, base64 version is generated.

generate_base64()
Generate the base64 representation from the generated thumbnail image file.

3.4.4 Utils module

ultimatethumb.utils.**get_cache_key**(*key*)
Generates a prefixed cache-key for ultimatethumb.

ultimatethumb.utils.**get_thumb_name**(*source*, ***options*)
Builds the thumbnail name and uses the name to store the source and options to cache.

ultimatethumb.utils.**get_thumb_data**(*thumb_name*)
Uses the thumbail name and fetches the source and options from cache.

ultimatethumb.utils.**parse_source**(*source*)
Parse and lookup the file system path for a given source. The function understands both media names and static names (if prefixed with "static:")

ultimatethumb.utils.**parse_sizes**(*value*)
Parses and returns a list of thumbnail sizes.

ultimatethumb.utils.**factor_size**(*value*, *factor*)
Factors the given thumbnail size. Understands both absolute dimensions and percentages.

ultimatethumb.utils.**get_size_for_path**(*path*)
Gets the image size for a given path. If the path does not exist, the call will fail loud with e.g. OSError exception.

ultimatethumb.utils.**get_domain_url**(*url*)
Returns the given url prefixed with a domain if configured in settings.

ultimatethumb.utils.**build_url**(*name*, *factor=1*)
Build the actual url for a given name and factor.

class ultimatethumb.utils.**MoveableNamedTemporaryFile**(*name*)
Bases: `object`

chunks()

close()

temporary_file_path()

3.4.5 Views module

class ultimatethumb.views.**ThumbnailView**(***kwargs*)

Bases: `django.views.generic.base.View`

ThumbnailView is used to provide the thumbnails to the brower.

We don't use a serve static view because we need to check if the requested thumbnail is available - and if not: generate one.

The view supports the X-Accel-Redirect header supported by Nginx to no really serve the binary (enabled by default if DEBUG is False).

get(**args*, ***kwargs*)

Fetch and return the thumbnail response.

get_thumbnail()

Try to fetch the thumbnail based on the thumbnail name. Might fail if cache resets between generation of the thumbnail urls and fetching of the images.

get_factor()

Get factor from url.

render_thumbnail(*thumbnail*, *factor*)

Generate the http response for the requested thumbnail. The code is able response with 304 if the thumbnail was already requested.

Supports X-Accel-Redirect, enabled by default if DEBUG is False.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

u

ultimatethumb.storage, 11
ultimatethumb.templatetags, 11
ultimatethumb.templatetags.ultimatethumb_tags, 11
ultimatethumb.thumbnail, 11
ultimatethumb.utils, 13
ultimatethumb.views, 14

INDEX

Symbols

`__init__()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`__init__()` (*ultimatethumb.thumbnail.ThumbnailSet method*), 11

B

`base64` (*ultimatethumb.thumbnail.Thumbnail property*), 12
`build_url()` (*in module ultimatethumb.utils*), 13

C

`chunks()` (*ultimatethumb.utils.MoveableNamedTemporaryFile method*), 13
`close()` (*ultimatethumb.utils.MoveableNamedTemporaryFile method*), 13

E

`exists()` (*ultimatethumb.thumbnail.Thumbnail method*), 12

F

`factor_size()` (*in module ultimatethumb.utils*), 13
`from_name()` (*ultimatethumb.thumbnail.Thumbnail class method*), 12

G

`generate()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`generate_base64()` (*ultimatethumb.thumbnail.Thumbnail method*), 13
`get()` (*ultimatethumb.views.ThumbnailView method*), 14
`get_base64_content()` (*ultimatethumb.thumbnail.Thumbnail method*), 13
`get_base64_path()` (*ultimatethumb.thumbnail.Thumbnail method*), 13
`get_cache_key()` (*in module ultimatethumb.utils*), 13
`get_domain_url()` (*in module ultimatethumb.utils*), 13
`get_estimated_size()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_factor()` (*ultimatethumb.views.ThumbnailView method*), 14
`get_gm_options()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_mimetype()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_name()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_size()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_size_for_path()` (*in module ultimatethumb.utils*), 13
`get_sizes()` (*ultimatethumb.thumbnail.ThumbnailSet method*), 11
`get_source_size()` (*ultimatethumb.thumbnail.ThumbnailSet method*), 11
`get_storage_name()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_storage_path()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_storage_url()` (*ultimatethumb.thumbnail.Thumbnail method*), 12
`get_thumb_data()` (*in module ultimatethumb.utils*), 13
`get_thumb_name()` (*in module ultimatethumb.utils*), 13
`get_thumbnail()` (*ultimatethumb.views.ThumbnailView method*), 14
`get_thumbnails()` (*ultimatethumb.thumbnail.ThumbnailSet method*), 11

H

`height` (*ultimatethumb.thumbnail.Size attribute*), 11

M

module
    ultimatethumb.storage, 11
    ultimatethumb.templatetags, 11
    ultimatethumb.templatetags.ultimatethumb_tags, 11
    ultimatethumb.thumbnail, 11
    ultimatethumb.utils, 13
    ultimatethumb.views, 14
MoveableNamedTemporaryFile (*class in ultimatethumb.utils*), 13

P

parse_sizes() (*in module ultimatethumb.utils*), 13
parse_source() (*in module ultimatethumb.utils*), 13

R

render_thumbnail() (*ultimatethumb.views.ThumbnailView method*), 14
requested_size (*ultimatethumb.thumbnail.Thumbnail property*), 12

S

Size (*class in ultimatethumb.thumbnail*), 11
size (*ultimatethumb.thumbnail.Thumbnail attribute*), 12

T

temporary_file_path() (*ultimatethumb.utils.MoveableNamedTemporaryFile method*), 13
Thumbnail (*class in ultimatethumb.thumbnail*), 12
ThumbnailFileSystemStorage (*class in ultimatethumb.storage*), 11
thumbnails (*ultimatethumb.thumbnail.ThumbnailSet attribute*), 11
ThumbnailSet (*class in ultimatethumb.thumbnail*), 11
ThumbnailStorage (*class in ultimatethumb.storage*), 11
ThumbnailView (*class in ultimatethumb.views*), 14

U

ultimatethumb() (*in module ultimatethumb.templatetags.ultimatethumb_tags*), 11
ultimatethumb.storage
    module, 11
ultimatethumb.templatetags
    module, 11
ultimatethumb.templatetags.ultimatethumb_tags
    module, 11
ultimatethumb.thumbnail
    module, 11
ultimatethumb.utils
    module, 13
ultimatethumb.views
    module, 14
url (*ultimatethumb.thumbnail.Thumbnail property*), 12
url_2x (*ultimatethumb.thumbnail.Thumbnail property*), 12

V

viewport (*ultimatethumb.thumbnail.Thumbnail attribute*), 12

W

width (*ultimatethumb.thumbnail.Size attribute*), 11