
Django-thumbnail-maker

Documentation

Release stable

March 08, 2015

| | | |
|----------|-------------------------------------|----------|
| 1 | How to Use | 3 |
| 1.1 | Get the code | 3 |
| 1.2 | Configure your project | 3 |
| 1.3 | Set up your model's field | 4 |
| 1.4 | Templates usage | 4 |
| 1.5 | Management commands usage | 4 |

Auto-generator of thumbnails for Django, using sorl-thumbnail.

- Auto-generates thumbnails using sorl-thumbnail while uploading (saving) images
- You can use any engines & plugins you usually use with sorl-thumbnail
- The application does not replace thumbnail templatetag and you can use everything you want from sorl-thumbnail
- Command to auto-generate missing thumbnails

How to Use

1.1 Get the code

Get the code for the latest stable release use pip:

```
$ pip install django-thumbnail-maker
```

1.2 Configure your project

Register `'thumbnail_maker'`, in the `INSTALLED_APPS` section of your project's settings:

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.admin',
    'django.contrib.sites',
    'django.contrib.comments',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.contenttypes',

    'thumbnail_maker',
)
```

Set up required thumb formats in `THUMBNAIL_MAKER_FORMATS` dictionary. Keys are the names for your formats (could be any string). Values are tuples of length 2: first element is a geometry string (used in `sorl-thumbnail`), second is a dictionary with options (`crop`, `quality`, `padding`, `format`, etc.):

```
THUMBNAIL_MAKER_FORMATS = {
    'banner': ('400x300', {'crop': 'center',
                          'quality': 90}),
    '50x50': ('50x50', {}),
    'any_name': ('5x277', {'padding': True})
}
```

You can set up `THUMBNAIL_MAKER_DEBUG` setting. By default it is set to `False` in order to pass exceptions while saving models or while generating batch of thumbnails. This setting is not required. Usage:

```
THUMBNAIL_MAKER_DEBUG = True
```

1.3 Set up your model's field

Use `ImageWithThumbnailsField` and `thumbs` option, where `thumbs` is a tuple of thumb format names (keys from `THUMBNAIL_MAKER_FORMATS` dictionary):

```
from django.db import models
from thumbnail_maker import ImageWithThumbnailsField

class Item(models.Model):
    image = ImageWithThumbnailsField(
        upload_to='somewhere',
        thumbs=('banner', '50x50'),
    )
```

1.4 Templates usage

All of the examples assume that you first load the `thumbnail_maker` template tag in your template:

```
{% load thumbnail_maker %}
```

A simple usage:

```
{% usethumbnail item.image "banner" as im %}
    
{% endusethumbnail %}

{% usethumbnail item.image "50x50" as im %}
    
{% endusethumbnail %}
```

You can also use string paths instead of image objects:

```
{% usethumbnail "dummy/image.png" "50x50" as im %}
    
{% endusethumbnail %}
```

1.5 Management commands usage

Django-thumbnail-maker comes with a `manage.py` command to generate missing thumbs. You can use it while

```
./manage.py make_thumbnails <app>.<model> <field>
```

In case you want to make all thumbs replacing old ones, use `--force` option:

```
./manage.py make_thumbnails --force <app>.<model> <field>
```