django-croppy Documentation

Release 0.1.0

Alen Mujezinovic

May 10, 2017

Contents

`django-croppy` enables creating custom crops of images by specifying a name, coordinates, width and height of the crop.

`django-croppy` provides a custom model field responsible for creating and deleting crops. Crops are stored as serialized JSON data on the same model as the image field via django-jsonfield.

`django-croppy` is useful if you want to manually curate the crop size and location instead of relying on generic cropping like django-imagekit provides.

`django-croppy` makes use of image processors provided by `django-imagekit`.

CHAPTER 1

Usage

First, create your model with a crop field. You can specify a custom location where to save crops to with the `upload_to` parameter:

```
from django.db import models
from croppy.fields import CropField

def upload_to(instance, filename, crop_name):
    """
    Default function to specify a location to save crops to.

    :param instance: The model instance this crop field belongs to.
    :param filename: The image's filename this crop field operates on.
    :param crop_name: The crop name used when :attr:`CropFieldDescriptor.crop` was
        called.
    """
    filename, ext = os.path.splitext(os.path.split(filename)[-1])
    return os.path.join('crops', u'%s-%s%s' % (filename, crop_name, ext))

class Image(models.Model):
    image = models.ImageField()
    crops = CropField('image', upload_to = upload_to)
```

The created `crops` field allows you to create, delete and inspect crops.

```
$ git clone git@github.com:caffeinehit/django-croppy.git
$ cd django-croppy
$ python manage.py syncdb
...
$ python manage.py shell
...
>>> from tests.app import tests
>>> image = tests.get_image('test.tiff')
>>> image
<Image: Image object>
>>> image.image
```

```
<ImageFieldFile: images/test.tiff>
>>> image.image.path
u'/home/alen/projects/django-croppy/tests/test-media/images/test.tiff'

>>> # Inspect the crop data
>>> image.crops.data
{}

>>> # Create a new crop called 'rect' at position 0/0
>>> # with a width of 100px and a height of 50px
>>> image.crops.create('rect', (0, 0, 100, 50))

>>> # Inspect the crop data
>>> image.crops.data
{'rect': {'y': 0, 'width': 100, 'height': 50, 'filename': 'crops/test-rect.tiff', 'x
↪': 0}}

>>> # Inspect the crop
>>> image.crops.rect.name
'crops/test-rect.tiff'
>>> image.crops.rect.path
u'/home/alen/projects/django-croppy/tests/test-media/crops/test-rect.tiff'
>>> image.crops.rect.url
'/test-media/crops/test-rect.tiff'

>>> # Save the data to database
>>> image.save()

>>> # Delete the crop
>>> image.crops.delete('rect')
>>> image.crops.data
{}
```

CHAPTER 2

API Documentation

croppy.fields