
PILKit Documentation

Release 2.0

Matthew Tretter

May 14, 2018

Contents

1	Installation	3
2	Usage Overview	5
2.1	Processors	5
2.2	Utilities	6
3	Community	7
4	Authors	9
4.1	Maintainers	9
4.2	Contributors	9
5	Contents	11

PILKit is a collection of utilities for working with PIL (the Python Imaging Library).

One of its main features is a set of **processors** which expose a simple interface for performing manipulations on PIL images.

Looking for more advanced processors? Check out [Instakit!](#)

For the complete documentation on the latest stable version of PILKit, see [PILKit on RTD](#).

build passing

CHAPTER 1

Installation

1. Install PIL or Pillow.
2. Run `pip install pilkit` (or clone the source and put the pilkit module on your path)

Note: If you've never seen Pillow before, it considers itself a more-frequently updated “friendly” fork of PIL that's compatible with setuptools. As such, it shares the same namespace as PIL does and is a drop-in replacement.

2.1 Processors

The “pilkit.processors” module contains several classes for processing PIL images, which provide an easy to understand API:

```
from pilkit.processors import ResizeToFit

img = Image.open('/path/to/my/image.png')
processor = ResizeToFit(100, 100)
new_img = processor.process(img)
```

A few of the included processors are:

- ResizeToFit
- ResizeToFill
- SmartResize
- Adjust
- TrimBorderColor
- Transpose

There’s also a ProcessorPipeline class for executing processors sequentially:

```
from pilkit.processors import ProcessorPipeline, ResizeToFit, Adjust

img = Image.open('/path/to/my/image.png')
processor = ProcessorPipeline([Adjust(color=0), ResizeToFit(100, 100)])
new_image = processor.process(img)
```

2.2 Utilities

In addition to the processors, PILKit contains a few utilities to ease the pain of working with PIL. Some examples:

prepare_image Prepares the image for saving to the provided format by doing some common-sense conversions, including preserving transparency and quantizing.

save_image Wraps PIL's `Image.save()` method in order to gracefully handle PIL's "Suspension not allowed here" errors, and (optionally) prepares the image using `prepare_image`

Utilities are also included for converting between formats, extensions, and mimetypes.

CHAPTER 3

Community

Please use the [GitHub issue tracker](#) to report bugs. A [mailing list](#) also exists to discuss the project and ask questions, as well as the official [#imagekit](#) channel on Freenode. (Both of these are shared with the [django-imagekit](#) project—from which PILKit spun off.)

4.1 Maintainers

- Bryan Veloso
- Matthew Tretter
- Chris Drackett
- Greg Newman
- Venelin Stoykov

4.2 Contributors

In addition to those listed on the [contributors page](#), the following people have also had a hand in bringing PILKit to life:

- Justin Driscoll
- Timothée Peignier
- Jan Sagemüller
- Alexander Bohn
- Eric Eldredge
- Germán M. Bravo
- Kevin Postal
- Madis Väin

CHAPTER 5

Contents

- genindex
- modindex
- search