
PyImageOptimizer Documentation

Release 1.0

Shantanu Khandelwal

May 18, 2016

1	PyImageOptimizer	3
1.1	Features	3
2	Installation	5
3	Usage	7
4	Contributing	9
4.1	Types of Contributions	9
4.2	Get Started!	10
4.3	Pull Request Guidelines	10
4.4	Tips	11
5	Credits	13
5.1	Development Lead	13
5.2	Contributors	13
6	History	15
7	1.0 (2015-01-11)	17
8	Indices and tables	19

Contents:

PyImageOptimizer

Python Image Optimizer . Uses jgpo.com and pngcrush.com to optimize the images.

- Free software: BSD license
- Documentation: <https://PyImageOptimizer.readthedocs.org>.

1.1 Features

- **TODO**
 - Expose Api to resize images

Installation

PyImageOptimizer requires Requests. Install that first with:

```
$ pip install requests
```

At the command line:

```
$ pip install PyImageOptimizer
```

or:

```
$ easy_install PyImageOptimizer
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv PyImageOptimizer  
$ pip install PyImageOptimizer
```

Usage

To use PyImageOptimizer in a project:

```
import PyImageOptimizer
```

To Just optimize the JPG images:

```
PyImageOptimizer.optimize_jpg(file_path, backup=False)
```

To Just optimize PNG images:

```
PyImageOptimizer.optimize_png(file_path, backup=False)
```

To optimise both jpg and png images (regardless of type):

```
PyImageOptimizer.optimize(file_path, backup=False)
```

file_path can be both a single file path and a path of a directory

changing backup to True will make a backup of your old file

Note:: PyImageOptimizer.optimize method uses ThreadPool , So if you have a large number of images try to use this method

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at <https://github.com/shantanu561993/PyImageOptimizer/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

4.1.4 Write Documentation

PyImageOptimizer could always use more documentation, whether as part of the official PyImageOptimizer docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/shantanu561993/PyImageOptimizer/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *PyImageOptimizer* for local development.

1. Fork the *PyImageOptimizer* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/PyImageOptimizer.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv PyImageOptimizer
$ cd PyImageOptimizer/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 PyImageOptimizer tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/shantanu561993/PyImageOptimizer/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_PyImageOptimizer
```

Credits

5.1 Development Lead

- Shantanu Khandelwal <shantanu561993@gmail.com>

5.2 Contributors

None yet. Why not be the first?

History

1.0 (2015-01-11)

- First release on PyPI.

Indices and tables

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