
Thumbnailer Documentation

Release 0.1dev

Rémy Hubscher

October 28, 2014

| | | |
|----------|------------------------------------|-----------|
| 1 | Contents | 3 |
| 1.1 | About Thumbnailer | 3 |
| 1.2 | Alternatives | 4 |
| 1.3 | Features | 4 |
| 1.4 | Architecture | 5 |
| 1.5 | Readers | 6 |
| 1.6 | Engines | 6 |
| 1.7 | Writers | 7 |
| 1.8 | Installation | 7 |
| 1.9 | Development guidelines | 9 |
| 1.10 | License | 13 |
| 1.11 | Authors and contributors | 15 |
| 2 | Indices and tables | 17 |

Thumbnailer is an ecosystem around thumbnails:

1. a thumbnail generation service, i.e. a web application providing an API;
2. tools to integrate the service in your favorite framework, i.e. template tag for Django;
3. various recipes to easily generate, host and serve thumbnails, i.e. server configuration samples and deployment scripts.

1.1 About Thumbnailer

This document presents the vision of the `Thumbnailer` project, as shared by *Authors and contributors*.

1.1.1 Why thumbnailer?

Thumbnail generation is a feature often used on websites. As developers, we **do not want**:

- to install and configure thumbnail-related modules, again and again, on every project we start.
- to use a different solution for every framework we use to build a website.
- to deal with deployment.
- thumbnail generation to downgrade our overall website performance.
- increase server's memory just to be able to perform operations on images.

Moreover, as internauts, we **do not want**:

- to manually edit images before we publish them.
- to install some image edition software just to create thumbnails.

That's why **we need a thumbnail generation service**.

Some *Alternatives* exist, but `Thumbnailer` has the following valuable features:

- Open-source. It can be used to create a hosted thumbnail service (saas), but you are also free to deploy it on your own infrastructure if you want.
- Extensible. Create, plug and configure *Engines*, *Writers* and *Readers*.
- Full ecosystem. If you want to manage your own thumbnail generation service, `Thumbnailer` provides ready-to-use recipes, including thumbnail generation, serving, caching...
- Generate thumbnails out of almost any document. Input can be images, PDF files, HTML pages, ...

However, this version of is still a proof of concept: all features aren't available yet.

1.1.2 Development status

`Thumbnailer` is under active development.

Currently, `Thumbnailer` project may contain parts of the ecosystem, so that it forms a product. Later, this project may be limited to the glue between ecosystem parts, these components being shipped as external projects.

1.2 Alternatives

This document lists services or projects that may provide similar functionality than the `Thumbnailer` project.

1.2.1 Services

- <http://www.webresourcesdepot.com/10-free-website-thumbnail-generation-services/>

1.2.2 Python packages

- <http://pypi.python.org/pypi/?%3Aaction=search&term=thumbnail&submit=search>
- <http://pypi.python.org/pypi/?%3Aaction=search&term=preview&submit=search>

1.3 Features

This document presents features of the `Thumbnailer` project.

As described in *About Thumbnailer*, the project is an ecosystem made of several components. So some features are, in fact, features of ecosystem parts.

1.3.1 Key features

- Open source;
- Simple but powerful API;
- Extensible;
- Easy to deploy.

1.3.2 Specifications

Feature: Get a thumbnail

To get a thumbnail of an image

As a user

I want to send access an url and get my thumbnail

Scenario: With an image to resize into a box

Given `/<engine>/?url=<url>&width=<width>&height=<height>`

When I access the api url

Then I get my image at max size `<width>x<height>`

Examples:

| | | | | |
|--------|---|-----|-----|--|
| engine | url | | | |
| scale | http://localhost:8000/images/horizontal.jpg | 800 | 600 | |
| scale | http://localhost:8000/images/vertical.jpg | 200 | 300 | |


```

| scale          | http://localhost:8000/images/vertical.jpg          | 200 | 0 |
| scale          | http://localhost:8000/images/vertical.jpg          | 0   | 200 |
| document       | http://localhost:8000/documents/document.pdf       | 500 | 500 |

```

Scenario: With an image to crop

Given /<engine>/?url=<url>&width=<width>&height=<height>

When I access the api url

Then I get my image at size <width>x<height>

Examples:

```

| engine | url                                     | width | height |
| crop   | http://localhost:8000/images/horizontal.jpg | 800   | 600   |
| crop   | http://localhost:8000/images/vertical.jpg   | 200   | 300   |
| upscale | http://localhost:8000/images/horizontal.jpg | 600   | 600   |
| upscale | http://localhost:8000/images/vertical.jpg   | 600   | 500   |

```

1.4 Architecture

This document presents typical architecture components of the Thumbnailer ecosystem.

1.4.1 Thumbnail generation service

This is Thumbnailer's core.

As input or configuration, the thumbnail generation service uses:

- a reader: gets the original resource and passes it as adequate input to the engine. Example: read an image identified by an URL. Learn more at *Readers*.
- an engine: transforms the input. Example: resize an image. Learn more at *Engines*.
- engine parameters: options that the engine understands. Example: width and height of the thumbnail.
- a writer: puts the result of the operation somewhere. Example: returns the thumbnail in an HTTP response. Learn more at *Writers*.

1.4.2 Thumbnails cache

Optional (but strongly recommended) component to improve performance and scalability.

If a thumbnail already exists in cache, serve it from cache.

When a new thumbnail is generated, store it in cache.

1.4.3 Asynchronous thumbnail generation

Optional components to be able to generate thumbnails asynchronously.

Provider

Emits requests of asynchronous thumbnail generation.

Broker

The brokers holds the queue of thumbnails to generate.

Worker

Consumes the broker's queue, communicates with the thumbnail generation service.

1.5 Readers

The thumbnail generation service uses readers to retrieve original documents. Readers return document in an adequate format for *Engines*.

Available readers:

- default reader: reads an image identified by an URL. Returns a PIL.Image ressource.

1.6 Engines

The thumbnail generation service uses engines to transform original documents into thumbnails. Engines take input from *Readers*, process them then pass result to *Writers*.

Engines also accept parameters. Typical parameters are width and height of the thumbnail.

1.6.1 Available engines

scale

Scale the input image to enter the box, if either width or height are empty, it will scale to fit provided value.

```
> curl -o thumb_scale.png 'http://localhost:5000/scale/?url=http://localhost:8000/images/horizontal.jpg'
< 200 OK + image/png thumb with max size 200x150
```

crop

Crop the input image at the right size.

```
> curl -o thumb_crop.png 'http://localhost:5000/crop/?url=http://localhost:8000/images/horizontal.jpg'
< 200 OK + image/png thumb with a center crop at size 200x150
```

upscale

Upscale the input image if it is too little for a crop.

```
> curl -o thumb_upscale.png 'http://localhost:5000/upscale/?url=http://localhost:8000/images/horizontal.jpg'
< 200 OK + image/png thumb with an upscale crop at size 200x150
```

document

Thumb a PDF file at the wanted size:

```
> curl -o thumb_pdf.png 'http://localhost:5000/document/?url=http://localhost:8000/document/document'
< 200 OK + image/png thumb with an upscale crop at max size 200x150
```

1.7 Writers

The thumbnail generation service uses writers to return thumbnails. Writers take result of *Engines* as input, and:

- actually put thumbnail content somewhere, typically in an HTTP response, but could be on some storage.
- return response to client, i.e. inform client of success or failure of the request.

Available writers:

- default writer: gets a PIL.Image and returns a HTTP response.

1.8 Installation

This document covers deployment of [Thumbnailer](#)¹ project.

1.8.1 OS specific

Here are recipes for specific operating systems. They should help you go fast or automate installation procedure.

Debian

The first lines in the following sh commands define some variables. Adapt them to your needs.

```
# Define some variables.
thumbnailer_dir=~/.thumbnailer # Installation directory.
thumbnailer_venv_dir=${thumbnailer_dir} # Virtualenv
upstream_url="git://github.com/Natim/Thumbnailer.git" # Main repository.
fork_url=${upstream_url} # Your fork.
system-install() { # Shortcut for system package installer.
  su -c "aptitude install --without-recommends ${*}";
}

# Install base system dependencies.
system-install git-core python-virtualenv
# Download project.
git clone ${fork_url} ${thumbnailer_dir}
# Create a virtualenv and activate it.
virtualenv ${thumbnailer_dir}
cd ${thumbnailer_dir}
source bin/activate
# Install core.
cd src/thumbnailer.core/
```

¹ <https://github.com/Natim/Thumbnailer>

```
python setup.py develop
cd ${thumbnailer_dir}
# Install images engine.
system-install libjpeg8 libjpeg8-dev libfreetype6 libfreetype6-dev
system-install python-dev
pip install PIL
cd src/thumbnailer.engines.images/
python setup.py develop
cd ${thumbnailer_dir}
# Install documents engine.
system-install rubygems graphicsmagick poppler-utils pdftk ghostscript
su -c "gem install docsplit"
pip install -U pip # Recent version of pip is required.
pip install git+https://github.com/anderser/pydocsplit@dev#egg=pydocsplit
cd src/thumbnailer.engines.documents/
python setup.py develop
cd ${thumbnailer_dir}

# Done!
# Run the server.
make provider_server
```

1.8.2 Generic guidelines

System requirements

Base

- Python² 2.7
- Git³

Provider (server)

- Flask⁴
- requests⁵

Images engine

- Python Imaging Library⁶ with JPEG and PNG support

Documents engine

- dev branch of pydocsplit⁷ and its dependencies

² <http://python.org/>

³ <http://git-scm.com/>

⁴ <http://flask.pocoo.org/>

⁵ <http://python-requests.org>

⁶ <http://www.pythonware.com/products/pil>

⁷ <https://github.com/anderser/pydocsplit/tree/dev/>

- Ghostscript ⁸ (see Graphicsmagicks add-on libraries installation notes ⁹)

Get the source

```
git clone git@github.com:Natim/Thumbnailer.git
```

Install Python packages

Install `thumbnailer.core`, `thumbnailer.engines.images` and `thumbnailer.engines.documents` in your Python environment.

You can use `setup.py` files provided at:

- `src/thumbnailer.core/setup.py`
- `src/thumbnailer.engines.images/setup.py`
- `src/thumbnailer.engines.documents/setup.py`

Run

Use the provided Makefile to run the server:

```
make runserver
```

By default, Thumbnailer's serves:

- `static/` directory on port 8000. Try `http://localhost:8000/`
- thumbnailer API on port 5000. Try `http://localhost:5000/scale/?url=http://localhost:8000/images/horizontal.jpg&width=100` and `http://localhost:5000/document/?url=http://localhost:8000/documents/document.pdf&width=400`

1.8.3 References

1.9 Development guidelines

This section is about contributing to the `Thumbnailer` project.

1.9.1 Table of contents

Install a development environment

Here are guidelines to get a development environment.

- You should create a ticket on [Thumbnailer's bugtracker](#) ¹⁰ **before** you fork and hack. Maybe someone already has a solution to your problem or feature request ;)
- Fork original repository if you plan to perform a pull request.
- Install `Thumbnailer`, as explained in *Installation*, except you use your fork's URL.

⁸ <http://www.ghostscript.com/>

⁹ <http://www.graphicsmagick.org/README.html#add-on-libraries-programs>

¹⁰ <https://github.com/Natim/Thumbnailer/issues>

- Install additional Python development tools:

- sphinx ¹¹
- lettuce ¹²

```
pip install sphinx lettuce
```

- Run tests:

```
make test
```

- Contribute:

- work in a separate branch, i.e. not in master. Prefix your branch name with the bugtracker's ticket number, so that we can identify it quickly.
- hack, test, commit and pull request...

Generic guidelines

Dependencies

- Python-2.7. You may use a virtual environment.

Install

```
# Download project from original repository... or use your own fork.
git clone https://github.com/Natim/Thumbnailer.git
cd Thumbnailer/
# Install Thumbnailer base with zc.buildout.
python lib/buildout/bootstrap.py --distribute
bin/buildout -N
# Install standard development tools.
bin/buildout -N install development
# That's all!
```

References

Testing

How to run and write tests for Thumbnailer.

Run tests

```
bin/lettuce
```

Contributing to the documentation

This document presents documentation conventions and tools.

This documentation uses Python-sphinx ¹³. It uses reStructuredText ¹⁴ syntax.

¹¹ <http://sphinx.pocoo.org/>

¹² <http://pypi.python.org/pypi/lettuce/>

¹³ <http://sphinx.pocoo.org/>

¹⁴ <http://docutils.sourceforge.net/rst.html>

Conventions

Language The documentation is written in english.

Line length Limit all lines to a maximum of 79 characters whenever possible. Exceptions can be long URL or some literal blocks.

Headings Follow the Sphinx's recommendation about sections ¹⁵.

As an example:

```
#####  
H1: document title  
#####
```

```
*****  
Sample H2  
*****
```

```
Sample H3  
=====
```

```
Sample H4  
-----
```

And sample text.

If you need more than H4, then consider creating a new document.

Code blocks Use the `code-block` directive and explicitly specify the programming language.

As an example:

```
.. code-block:: python  
  
    import this
```

Links and references On pages which are quite long, use links and references footnotes with the “target-notes” directive. As an example:

```
#####  
Some document  
#####
```

Some text which includes links to ``Example website`_` and many other links.

``Example website`_` can be referenced multiple times.

(... document content...)

And at the end of the document...

```
*****  
References
```

¹⁵ <http://sphinx.pocoo.org/rest.html#sections>

```
*****
```

```
.. target-notes::
```

```
.. _`Example website`: http://www.example.com/
```

This *Contributing to the documentation* page uses this syntax.

Install Sphinx

Python-sphinx ¹ installation is covered in *Install a development environment*.

In other cases, please refer to Python-sphinx ¹ documentation.

Export documentation to HTML

Go to docs/ folder in Thumbnailer project and use the provided Makefile:

```
cd docs/  
make html  
cd ..
```

HTML documentation is exported to docs/_build/html/.

Doctests

This documentation uses the Sphinx's doctest extension ¹⁶.

Write doctests Here is a RST code sample to write doctests.

```
.. doctest::
```

```
>>> print "Hello world!"  
Hello world!
```

See Sphinx's doctest extension ⁴ and Python's doctest ¹⁷ documentations for details.

Run doctests Go to docs/ folder in Thumbnailer project and use the provided Makefile:

```
cd docs/  
make doctests  
cd ..
```

¹⁶ <http://sphinx.pocoo.org/ext/doctest.html#module-sphinx.ext.doctest>

¹⁷ <http://docs.python.org/library/doctest.html>

References

1.10 License

1.10.1 GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>> Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, “this License” refers to version 3 of the GNU Lesser General Public License, and the “GNU GPL” refers to version 3 of the GNU General Public License.

“The Library” refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An “Application” is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A “Combined Work” is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the “Linked Version”.

The “Minimal Corresponding Source” for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

1. under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or
2. under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

1. Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License.
2. Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work and reverse engineering for debugging such modifications, if you also do each of the following:

1. Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License.
2. Accompany the Combined Work with a copy of the GNU GPL and this license document.
3. For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document.
4. Do one of the following:
 - (a) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.
 - (b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version.
0. Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

1. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.
2. Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy’s public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

1.11 Authors and contributors

- Rémy Hubscher <remy.hubscher@ionyse.com>
- Benoît Bryon <benoit@marmelune.net>

Indices and tables

- *genindex*
- *modindex*
- *search*