latexrender Documentation

Release 0.3.6

Luke Pomfrey

December 24, 2013

Contents

Contents:

CHAPTER 1

latexrender

A simple Flask app for rendering LaTeX snippets into images using `latex` and `dvipng`.

- Free software: BSD license
- Documentation: http://latexrender.rtfd.org.

CHAPTER 2

Installation

At the command line:

```
$ easy_install latexrender
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv latexrender
$ pip install latexrender
```

You'll also need the `latex` and `dvipng` binaries installed on your system.

CHAPTER 3

Usage

3.1 Configuration

Setup is done using the following environment variables:

Variable	Notes	Default
LATEXRENDER_OUTPUT_DIR	Directory where image files will be saved and served from	`/tmp/latexrender/`
LATEXRENDER_TEMPLATE	Jinja2 template to use for rendering the LaTeX snippet.	`template.tex` in installation directory
LATEXRENDER_XELATEX	Location of xelatex executable.	(Auto)
LATEXRENDER_DVIPNG	Location of dvipng executable.	(Auto)
USE_X_SENDFILE	Whether to use X-Sendfile or serve the generated image directly.	`True`
LATEXRENDER_SENDFILE_ROOT	A root directory to prepended to the X-Sendfile header sent to the upstream proxy.	Empty string

If not specified your path will be inspected to find `latex` and `dvipng`.

The default template, in Jinja2 format, is

```
{# vim: set ft=jinja ts=4 sw=4 tw=0 et :#}
\documentclass[12pt]{article}
\usepackage[utf8]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{color}
\usepackage{cancel}
\pagestyle{empty}
\begin{document}
\huge
{{ latex }}
\end{document}
```

3.2 Deploying

Run using your favourite WSGI server:

```
chausette latexrender.app
gunicorn latexrender:app
```

It is recommended that you run this behind a frontend proxy such as nginx and use X-Sendfile to have that serve the image files.

Pass base64 encoded LaTeX snippets as the URL in one of the following forms:

```
http://localhost:8080/<b64latex>/
http://localhost:8080/<b64latex>.png
```

In the default template the surrounding math environment is not specified so you can, for example, use input of the form:

```
$$ e^{i\pi} + 1 = 0 $$
```

or:

```
\begin{equation*}
    e^{i\pi} + 1 = 0
\end{equation}
```

CHAPTER 4

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/lpomfrey/latexrender/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

latexrender could always use more documentation, whether as part of the official latexrender 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/lpomfrey/latexrender/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 *latexrender* for local development.

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

   ```
   $ git clone git@github.com:your_name_here/latexrender.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 latexrender
   $ cd latexrender/
   $ 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 latexrender 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, and 3.3, and for PyPy. Check https://travis-ci.org/lpomfrey/latexrender/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_latexrender
```

CHAPTER 5

Credits

5.1 Development Lead

- Luke Pomfrey <lpomfrey@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

Indices and tables

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