PyResume Builder Documentation

Release 0.1.3

Wayne Warren

Sep 21, 2020

CONTENTS

1 PyResume Builder **3**
- 1.1 Features . . . 3
- 1.2 Similar Projects . . . 4
- 1.3 Credits . . . 4

2 Installation **5**
- 2.1 Requirements . . . 5
- 2.2 Stable release . . . 5
- 2.3 From sources . . . 5

3 Usage **7**
- 3.1 Basic . . . 9
- 3.2 Advanced . . . 9
- 3.3 Tips & Tricks . . . 10

4 Contributing **11**
- 4.1 Types of Contributions . . . 11
- 4.2 Get Started! . . . 12
- 4.3 Pull Request Guidelines . . . 13
- 4.4 Testing Tips . . . 13

5 Authors **15**
- 5.1 Development Lead . . . 15
- 5.2 Contributors . . . 15

6 History **17**
- 6.1 0.1.3 (2018-06-11) . . . 17
- 6.2 0.1.2 (2018-06-10) . . . 17
- 6.3 0.1.1 (2018-06-09) . . . 17
- 6.4 0.0.0 (2017-06-01) . . . 17

7 Indices and tables **19**

Contents:

CHAPTER

ONE

PYRESUME BUILDER

Do you like updating your resume? Are you satisfied with the layout and formatting? Do you ever wish you could try different styles without going through the trouble of manually reformatting each time? Well now you can!

PyResume Builder is a command line tool that populates a LaTeX template from contents defined in a YAML file. By storing the contents of your resume in a human-readable markup language and generating a LaTeX or PDF file from that you are free to try alternative templates and generate resumes in different formats without the tedium of manually reformatting.

- Documentation: https://pyresume.readthedocs.io.

1.1 Features

1.1.1 Current

- Store resume contents (skills, experience, contact info, etc) in a YAML file for ease of updating and version control convenience.
- Templated LaTeX approach allows for consistent look and feel between different combinations of information you might want to include in your resume.
- Default LaTeX template includes support for:
  - Contact info
  - Education
  - Experience
  - Skills (up to two levels of subcategories supported)
  - Activities
  - Education
  - References

1.1.2 Planned

- Support externally-defined Jinja2 LaTeX templates.
- Support some kind of html output format.
- Create new resume templates using cookiecutter.

1.2 Similar Projects

http://www.resumebuilder.org This site let's you sign in, enter your resume data in a series of web forms, and generate a resume from predefined templates. Also seems to include all kinds of helpful advice about resume layouts and being more of a rock star potential employee.

1.3 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

CHAPTER

TWO

INSTALLATION

2.1 Requirements

Python >= 3.5 The only tested version of python so far.

2.1.1 For Generating PDFs

One of the following tools should be sufficient to actually generate PDFs:

docker Provides full texlive installation for the sake of producing PDF output in tests. In the future there will be a `pyresume create pdf` command that uses this image to generate PDFs directly for users.

texlive (or some other LaTeX tool) A relatively minimal installation should take care of most resume-generating needs.

Installation steps for either of these will vary depending on the operating system (ie windows, mac, linux, etc); in the case of linux, your OS's package management tool (yum, apt, nix, pacman, emerge, etc); and on your OS version. However, I would recommend trying docker first since that will open up other avenues of computer usage for you.

2.2 Stable release

To install PyResume Builder, run this command in your terminal:

```
$ pip install pyresume
```

This is the preferred method to install PyResume Builder, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.3 From sources

The sources for PyResume Builder can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/waynr/pyresume
```

Or download the tarball:

```
$ curl  -OL https://github.com/waynr/pyresume/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER

THREE

USAGE

PyResume Builder is a command line tool that takes a path to a YAML file containing resume data and produces a LaTeX document. The following is an example of YAML that will produce a resume using all features of the basic template.

```yaml
basic:
  name: McMeow
  address:
    - 1537 Paper St
  contact:
    email: meow@meow.meow
    phone: 867 5309
  websites:
    - text: Cats-R-Us
      url: https://meow.meow/meow
    - text: Cats-N-Computers
      url: https://miaow.miaow/miaow

objective: >
  I am currently looking for someone to pay me to meow.

education:
  - school: Southern Meowniversity
    startdate: Fall 2005
    enddate: Spring 2009
    degrees:
      - Bachelor of Meow, Computer Meongineering
      - "Minor, Mewthematics \\& Bit Meowta"
    achievements:
      - "Dean's Cat List"
    gpa: 3.meow/4.0

experience:
  - company: Meow Meow Inc
    titles:
      - name: Lead Meower
        startdate: April 2016
        enddate: May 2017
    projects:
      - "Tech lead on project involving meowing until food appears."
      - "Mrowled at the moon."

  - company: Mewling Daycare
    titles:
      - name: Quality Eng.
```

(continues on next page)

(continued from previous page)

```
        startdate: March 2014
        enddate: March 2016
    projects:
      - "Designated mew troubleshooter."
      - "Trained 5994 kittens in litterbox usage."

activities:
  - name: "Meowing Endlessly"
    startdate: "Fall 2016"
    enddate: "Spring 2017"
  - name: "Core Meower's Team"
    startdate: "Fall 2010"
    enddate: "Fall 2010"

skills:
  - category: Caterwauling Languages
    skills:
      - category: Advanced
        skills:
          - Mew
          - Pur
      - category: Intermediate
        skills:
          - Meow
          - Mrow
      - category: Beginner
        skills:
          - Miaow
          - Meao
  - category: Cat Skills
    skills:
      - meowing
      - sleeping
      - eating
      - pooping
      - chasing insects
```

Before you begin, please be sure to read *installation instructions*. If you have any trouble, please be sure that all the *Requirements* have been satisfied. Also consider *submitting feedback* if you encounter any issues (bug, problem with docs, deepfelt yearning for existential meaning)

You will have the best chance of successfully using `pyresume` if you are running either Linux or Mac.

3.1 Basic

Basic usage is basic because it assumes that all you care about is generating LaTeX. If you want to go further and produce a PDF or some other output format supported by LaTeX, see the Advanced section below.

3.1.1 Single Yaml -> LaTeX

The following command assumes you have a file called `./attributes.yaml` in the directory in which you are running commands. If you don't, feel free to copy the contents of the full example shown above into a new file called `./attributes.yaml`.

```
$ pyresume create tex ./attributes.yaml > my-resume.tex
```

The result is a file called `my-resume.tex`. You can open this file with a text editor, but it will look like gibberish unless you are familiar with TeX, LaTeX, or etc. To quote https://www-project.org/about/ ,

> LaTeX, which is pronounced «Lah-tech» or «Lay-tech» (to rhyme with «blech» or «Bertolt Brecht»), is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing.

Most people may not care about that so it will be left to the reader to decide whether to **`learn more about LaTeX<https://latex-project.org>`_**. (You may need to learn more if you aren't running a linux distro with either `docker` or some kind of software that can read LaTeX and spit out PDFs.)

3.1.2 Multi Yaml -> LaTeX

What if you want to keep different types of attributes in their own file? See below, where the `./basic.yaml` contains all the contact information that you might (for example) want to keep out of a git repository, or in a more private git repository.

```
$ pyresume create tex ./attributes.yaml ./basic.yaml > my-resume.tex
```

As with the single yaml to LaTeX use case, the result here is a file called `my-resume.tex`.

3.2 Advanced

These are considered "advanced" usage because it involves using programs other than `pyresume` itself.

3.2.1 Generate PDF Using Docker Image

This usage assumes that you have docker installed. If you don't have it installed, then you should figure that out before continuing. The docker website has some **`good documentation for getting started<https://docs.docker.com/get-started/>`_**.

The following commands will run a docker container that uses an image known to have a complete installation of `texlive`. Note that this assumes you have an `attributes.yaml` file in your current working directory:

```
$ pyresume create tex /path/to/attributes.yaml > my-resume.tex
$ docker run --rm -it -v $PWD:/doc/ thomasweise/texlive pdflatex.sh /doc/my-resume.tex
```

This should result in a `my-resume.pdf` file in your current working directory. If it doesn't, then please feel free to *Report Bugs*!

3.2.2 Generate PDF on Linux Using `texlive`

These instructions should vary depending on the specific distro you are using. If your distro is not listed here but you happen to know how to adapt these instructions for it, please consider *contributing documentation*.

Debian 9/Stretch

You should install and `texlive` and `latexmk` if you haven't already:

```
$ sudo apt-get install texlive latexmk
```

Once you have, the following will create your PDF resume:

```
$ pyresume create tex /path/to/attributes.yaml > my-resume.tex
$ latexmk -pdf my-resume.tex
```

This should result in a `my-resume.pdf` file in your current working directory. If it doesn't, then please feel free to *Report Bugs*!

3.3 Tips & Tricks

3.3.1 Version Control

Create a git repository to version-control your resume:

```
$ mkdir -p /home/me/projects/personal-resume
$ cd /home/me/projects/personal-resume

$ git init

$ $EDITOR attributes.yaml # Fill it with all your secrets.
$ git add attributes.yaml
$ git commit -m "My very first version-controlled resume."
```

If you also consider pushing it to a public git repo be careful not to add personal contact info that you wouldn't want everyone to see! One way to avoid this is to keep your "basic" section out of the bulk of your resume attributes, like so:

```
$ $EDITOR attributes.yaml # Fill it your skills, education, work history, etc
$ $EDITOR basic.yaml # Fill it your pyresume "basic" section.
$ git add attributes.yaml
$ echo "basic.yaml" >> .gitignore
$ git add .gitignore
$ git commit -m "My very first version-controlled resume."
```

Have your own tips & tricks? Consider *writing some documentation*!

CHAPTER

FOUR

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/waynr/pyresume/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" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

4.1.4 Write Documentation

PyResume Builder could always use more documentation, whether as part of the official PyResume Builder 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/waynr/pyresume/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 *pyresume* for local development.

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

```
$ git clone git@github.com:your_name_here/pyresume.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 pyresume
$ cd pyresume/
$ 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've made your changes to pyresume source, run the tests:

```
 $ tox -e py37

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

6. When you're done making changes to code and tests, please format your code (CI tests will fail if you don't):

```
 $ tox -e black-fmt

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

7. 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
```

8. 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 3.5. Check https://travis-ci.org/waynr/pyresume/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Testing Tips

4.4.1 Run a Subset of tests

4.4.2 Generate New Test Fixtures

To generate fixtures that use the standard/default resume template

```
$ mkdir /path/to/pyresume/tests/scenarios/fixtures/standard/<new-scenario>/
$ $EDITOR /path/to/pyresume/tests/scenarios/fixtures/standard/<new-scenario>/
↪attributes.yaml
$ pyresume create tex > /path/to/pyresume/tests/scenarios/fixtures/standard/<new-
↪scenario>/attributes.tex
```

You can validate this works as expected by running the tests

```
$ py.test tests.scenarios.test_scenarios
```

Alternatively, you could just run the entire test suite

```
$ tox -e py37
```

Assuming there new resume templates are eventually added, creating the fixture might look something like:

```
$ mkdir /path/to/pyresume/tests/scenarios/fixtures/<new-template-name>/<new-scenario>/
$ $EDITOR /path/to/pyresume/tests/scenarios/fixtures/<new-template-name>/<new-
↪scenario>/attributes.yaml
$ pyresume create --template <new-template-name> tex > /path/to/pyresume/tests/
↪scenarios/fixtures/<new-template-name>/<new-scenario>/attributes.tex
```

And of course you will want to commit these to the git repo

```
$ git add /path/to/pyresume/tests/scenarios/fixtures/<new-template-name>/<new-
↪scenario>/
```

CHAPTER

FIVE

AUTHORS

5.1 Development Lead

- Wayne Warren <wayne.warren.s@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER

SIX

HISTORY

6.1 0.1.3 (2018-06-11)

- Fix packaging and testing with src/ layout.

6.2 0.1.2 (2018-06-10)

- Fix runtime dependencies.
- Change encrypted pypi password in .travis.yml.

6.3 0.1.1 (2018-06-09)

- Fix packaging bug - https://github.com/waynr/pyresume/issues/95

6.4 0.0.0 (2017-06-01)

- First release on PyPI.

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search