Tips for making a Publicly Available Code

Intro to Python - July 12, 2022

Why make code publicly available?

- Reproducibility
- Publicizing your work and building a CV/resume
- Code developed under NASA funding is now required to be made open-source

Three things to know/learn:

- Github
- PyPI
- Read the Docs

Github essentials

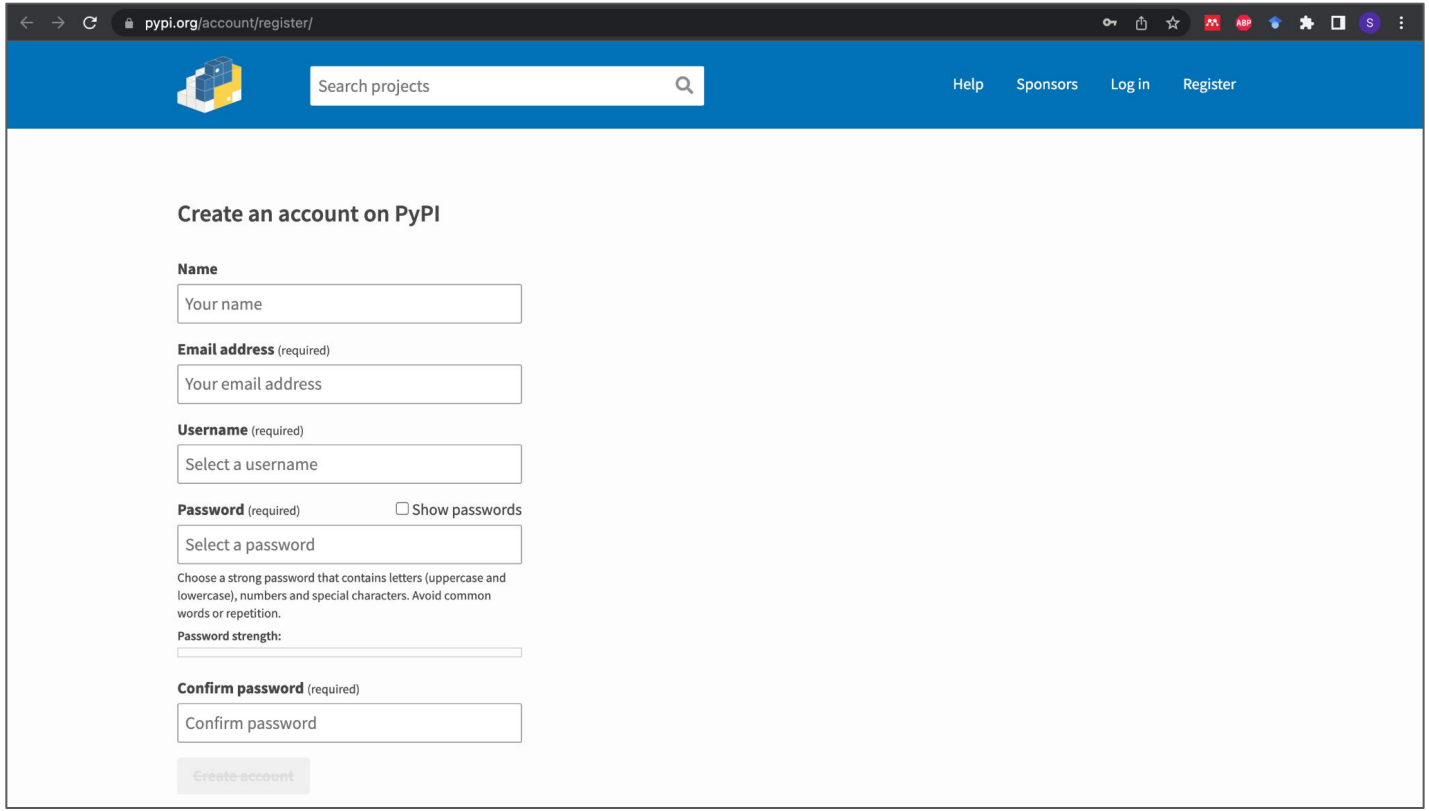
- Your code (duh)
- License
  - Typically people use the MIT license, which permits anybody to copy and use your code.
- Readme
  - Including details such as how to install the package and how to attribute the code.
- A setup.py file
  - Required for making your code installable.
  - Will say more about this later.
- Example notebooks showing how to use the code

https://github.com/stevengiacalone/triceratops

PyPI

- PyPI is a “repository of software for the Python programming language.”
- Uploading your code to the PyPI website makes your code pip installable, and therefore more accessible to others.
- How do you use PyPI?
  - The following tutorial can be found here: https://dzone.com/articles/executable-package-pip-install

PyPI Step 1 - Register on the PyPI website



PyPI Step 2 - Install required packages

- **Setuptools:** Setuptools is a package development process library designed for creating and distributing Python packages.
- **Wheel**: The Wheel package provides a `bdist_wheel` command for `setuptools`. It creates .whl file which is directly installable through the `pip install` command. We'll then upload the same file to pypi.org.
- **Twine**: The Twine package provides a secure, authenticated, and verified connection between your system and PyPi over HTTPS.
- **Tqdm**: This is a smart progress meter used internally by Twine.

PyPI Step 3 - Make a setup.py file and a license

Meta-Data	Description
name	Name of your package.
version	Current version of your pip package.
scripts	List of executable files. It's recommended to keep them the same as your pip package name. Here we are using `dokr`.
author and author_email	Name and Email Id of the author.
description	A short description of the package.
long_description	A description of the package.
long_description_content_type	A longer description. Here it is markdown. We are picking README.md for the long description.
packages	Use for other package dependencies.
classifiers	Contains all the classifiers of your project.

PyPI Step 3 - Make a setup.py file and a license

```python
from setuptools import setup, find_packages

def readme():
    with open('README.rst') as f:
        return f.read()

setup(name = "triceratops",
      version = '1.0.16',
      description = "Statistical Validation of Transiting Planet Candidates",
      long_description = readme(),
      author = "Steven Giacalone",
      author_email = "steven_giacalone@berkeley.edu",
      url = "https://github.com/stevengiacalone/triceratops",
      packages = find_packages(),
      package_data = {'triceratops': ['data/*']},
      classifiers=[
        'Development Status :: 5 - Production/Stable',
        'Intended Audience :: Science/Research',
        'Operating System :: OS Independent',
        'Programming Language :: Python :: 3',
        'License :: OSI Approved :: MIT License',
        'Topic :: Scientific/Engineering :: Astronomy'
        ],
      install_requires=['numpy>=1.18.1','pandas>=0.23.4', 'scipy>=1.1.0', 'matplotlib>=3.5.1',
                        'astropy>=4.0', 'astroquery>=0.4.6', 'pytransit>=2.2',
                        'mechanicalsoup>=0.12.0', 'emcee>=3.0.2', 'seaborn>=0.11.1',
                        'numba>=0.52.0', 'pyrr>=0.10.3', 'celerite>=0.4.0', 'lightkurve>=2.0.0'],
      zip_safe=False
)
```

PyPI Step 4 - Compile your package

- Run “python setup.py bdist_wheel”
- This will create 3 directories in your repository:
  - **build/**
  - **dist/**
  - **project.egg.info/**
- You can now install the package on your local machine to test it. I recommend doing this before uploading to PyPI, so that you don’t have to constantly yank and re-upload new releases.
  - Do this by running “python -m pip install dist/package.whl”

PyPI Step 5 - Upload your code on pip

- Once you are happy with your package, run the following:
  - “python -m twine upload dist/*”
- It will ask you to provide your PyPI username and password.
- After entering these, your package will be pip installable and will have a page on PyPI.
- Example: https://pypi.org/project/triceratops/

Read the Docs

- Read the Docs is a convenient place to keep info about your package, tutorials, and APIs.
- Some good examples:
  - exoplanet by Dan Foreman-Mackey: https://docs.exoplanet.codes/en/latest/
  - batman by Laura Kreidberg: http://lkreidberg.github.io/batman/docs/html/index.html
  - starry by Rodrigo Luger: https://starry.readthedocs.io/en/latest/
- Read the Docs pages are made using sphinx. You can find a tutorial here: https://sphinx-tutorial.readthedocs.io/start/

RTD Step 1: Getting started with sphinx

- pip install sphinx
- Clone the tutorial repository: “git clone https://github.com/ericholscher/pycon-sphinx-tutorial”
- cd into the project directory (“crawler” in this case), make a “docs” directory and cd into it.
- Run “sphinx-quickstart” and answer the prompts that pop up.
- Build the docs into html by running “make html” in the “docs” directory.
- You can open your docs by running: “open _build/html/index.html”

RTD Step 2: Customize your theme

- Install the standard RTD theme: “pip install sphinx_rtd_theme”
- Go to your conf.py file and put the following:

```
import sphinx_rtd_theme

html_theme = 'sphinx_rtd_theme'

html_theme_path = [sphinx_rtd_theme.get_html_theme_path()]
```

- Re-run “make html” to update your doc pages.
- Other themes can be found here: https://sphinx-themes.org/

RTD Step 3: Create your doc pages

- Edit the homepage and structure of your docs in your index.rst file.
- Let's run through an example.
- Pages are usually .rst or .md files, but you can also render jupyter notebooks in your RTD: https://docs.readthedocs.io/en/stable/guides/jupyter.html
  - Do do this, you must first pip install nbsphinx
  - import nbsphinx in conf.py and add "nbsphinx" to extensions