PyPi Documentation

Thibaut Meric

12.03.2015

Table of Contents

Part I
Introduction

1 Purpose of this document

This documentation is meant to help its readers releasing a Python program thru PyPi. Once a Python project is over, handling its release to a large scale, and ensure a version control can be a problem. Using PyPi solve those two issues.

2 What is PyPi

PyPi can be considered as an application store for Python programs.

- As a user, in commandline, you can download programs directly from PyPi, using pip command. Just type in command prompt:

  ```
  pip MyPackage
  ```

  And if MyPackage is on PyPi, it will be installed.

- As a developer, you can push any python program you want, with no restrictions and no control -as soon as the name of your package is not already used obviously-.

Part II
Installation

1 Install Python

The first step consist in verifying if Python is correctly installed on your computer.

Note: By default Python should be installed on MacOs.

In the command prompt, to check if Python is installed, type:

```
Python -V                                   (For Mac & Linux)
py                                          (For Windows)
```

If this command returns an error, you need to install Python. Go to https://www.python.org/downloads/ and install Python 3.xx.

2 Install pip

Note: At this point, for Windows users, you need to link your command prompt with the new installed scripts, if you don't know how to do that, go to chapter 4.1.

Theoretically, pip comes with Python 3.4 or higher. To Check if pip is installed, type:

```
pip
```

If this command returns an error you need to install pip. Go to https://bootstrap.pypa.io and download "get-pip.py".
Once downloaded, double click on the file. Pip should be installed automatically.

Note: For Mac users, you may need to run this file thru your command prompt. Just place the command prompt in the same folder than your file and type:
python get-pip.py.
If you don't know how to move into the file system in commandline, go to chapter 4.2

3 Install wheel

Know, pip should be installed on your computer. PyPi supports several installation packages types, but the most commons are .egg files and .whl files. whl files tends to become the new standard, and is recommended by PyPi, so we are only going to talk about .whl.
To create new .whl files, wheel package has to be installed. Just type:

```
pip list
```

to check if wheel is installed. If not, install it using pip:

```
pip install wheel
```

Note: Wheel doesn't have to be installed on the user side to unpack your .whl file

4 Install twine

Twine is a package made to interact with PyPi. It's not the only available utility to talk with PyPi, but this one use HTTPS protocol. Just type:

```
pip list
```

to check if twine is installed. If not, install it using pip:

```
pip install twine pyopenssl ndg-httpsclient pyasn1
```

Part III
Working with PyPi

1 Create a .whl file

1.1 Create a setup.py file

To create a whl file, wheel needs a setup file named "setup.py". This file has to be placed at the top level of your project. This setup file will be used to reference the files you want to appear in your project, the name of your whl file, and general information around your project.

Here is asetup.py example:

```
from setuptools import setup, find_packages
import MainFolder

    setup(
        name = 'MyProjectName',
        version = MainFolder.__version__,
        packages = find_packages(),
        entry_points = 'gui_scripts': ['CalledNameInCommandPrompt = MainFolder.MyMainFile:main',],
        description='example',
        long_description='This is a long description for an example',
        author='Me',
        author_email='me@microchip.com',
        url='www.xample.com',
        maintainer= 'Not Me',
        maintainer_email='not.me@microchip.com',
        classifiers=['Development Status :: 5 - Production/Stable',
            'Intended Audience :: Developers'
            'License :: OSI Approved :: MIT License',
            'Natural Language :: English',
            'Programming Language :: Python :: 2.3',
            'Programming Language :: Python :: 3',
            ],
        platforms=['Operating System :: MacOS :: MacOS X','Operating System :: Microsoft :: Windows'],
        keywords='Example',
    )
```

The setup file will scan the directories entered in the 'packages" attribute, and add them to the whl file.

Caution: In each folder of the project, an empty __init__.py file must be present. Or the setup scan will not include the entire folder and subdirectories to the whl file.

Note: All documentation about setup() attributes can be found here:
https://packaging.python.org/en/latest/distributing/#setup-args

Note: All classifiers attributes can be found here:
https://pypi.python.org/pypi?%3Aaction=list_classifiers
All those attributes will be used by PyPi to reference your project and name the .whl file.
The name of your whl file is really important. It contains information that, for example will determine if you can install the package or not, depending on the version of python you have.

All documentation around wheel file name can be found here:
https://www.python.org/dev/peps/pep-0427/

PyPi is also really strict on version number. The version number is always inserted in the name of the whl file, and it is not possible to submit two times with the same version number.

1.2 Create a universal .whl file

By default, your program can only work either on Python 2 or Python 3, but not on both at the same time. If you want your wheel to work on both environments, building a "universal" wheel is needed.

To do so, create a file called "setup.cfg" is the same directory than "setup.py" and type in the file:

```
[bdist_wheel]
universal = 1
```

Save and close
Now, once we will generate our .whl file, our wheel will be a universal wheel, working either on Python 2 and Python 3.

Note: More information about wheel type can be found here:
http://python-packaging-user-guide.readthedocs.org/en/latest/distributing/#wheels

1.3 Create a .whl file

Once the setup.py file created, we can create our .whl file.

To test the setup.py file, go with your command prompt into the setup file directory, and type:

```
python setup.py test
```

Note: If you don't know how to move into the file system in commandline, go to chapter 4.2

If it returns no error, we can go to the next step. To create a whl file type:

```
python setup.py bdist_wheel
```

And the whl file be generated in a folder named dist.

1.4 Locally test your .whl file

To test the new whl file, go into dist directory with your command prompt, and type:

```
pip install MyProjectName-1.0-py2-none-any.whl
```

Once installed run the program typing in comand prompt the entry_points name you set up in script.py. In our previous example this name was: "CalledNameInCommandPrompt".

If you want to delete this new installation just type:

```
pip uninstall MyProjectName
```

2 Submit your .whl to PyPi

To submit your program to PyPi, the only requirement is to have a free account on PyPi. It is recommended to create an account directly on their website, but you can also do it in commandline.

Note: PyPi also has a test website to test your program before doing an official release. Its adress is https://testpypi.python.org/pypi. To use it, you have to create a separate account from the main PyPi website.
We need to submit two elements to make a clean release.
First, we need to submit the script.py file, that will contain all the information about our whl file.

To do so, go into the setup.py directory and type:

```
python setup.py register
```

If you don't have an account, you can create one at this step, but it's not recommended.
Once done, we can know submit our whl file.

To do so, just type:

```
twine upload dist/MyProjectName-1.0-py2-none-any.whl
```

Your Program is now available for everyone. If you want to work around your new release, just go on https://pypi.python.org, log on and edit your release.

3 Download your .whl from PyPi

if you want to download and install your whl file, just type:

```
pip install MyProjectName
```

Part IV
Extra

1 Add commands to windows command prompt

Window command prompt works with environment variables. Those environment variables are stored in .exe files placed anywhere on the computer. Windows will store the paths of those .exe files in a file to get acess to them. Once you type a command, the command prompt will try each .exe file until it find yours. Pip on windows is a .exe file stored in the Script folder in Python installation diretory. To tell windows this .exe file has to be added to the environment variable, we need to edit the path file. To do so, type in windows command prompt:

```
setx PATH "%PATH%;C:\Path\To\Python\Install\Scripts"
```

Where:

- PATH is the file where the paths are stored
- %PATH% represent all the old reference that we keep
- ; a spacer between old and new path
- C:\Path\To\Python\Install\Scripts is the path of your script folder in the Python directory. You shall only edit this part of the commandline and put your path. If you didn't change anything in Python installation paths, this path should be: C:\PythonXX\Scripts

Note: After this, you need to restart the command prompt, to apply changes.

2 Move into file system with command prompt

When you type a command in the command prompt, this command is always executed from a directory. That means that when you want to use commands dealing with files, the path to these files will have to be from the current directory to the file, and NOT the absolute path of these files.
If you want to know in which directory your command prompt is, type:

```
pwd                                   (For Mac and Linux)
echo %cd%                             (For Windows)
```

If you want to change the current directory, use:

```
cd ..                                 (To go to the parent folder)
cd MyFolder                           (To go inside MyFolder)
```

If you want to know what folders and files are in your curent directory just type:

```
ls                                    (For Mac & Linux)
dir                                   (For Windows)
```