Python Package Guide

While learning python, when we reach some level, where we can build some complex programs on our own!

Some thoughts evoke us that "Can I put my code on python libraries like other pip modules"

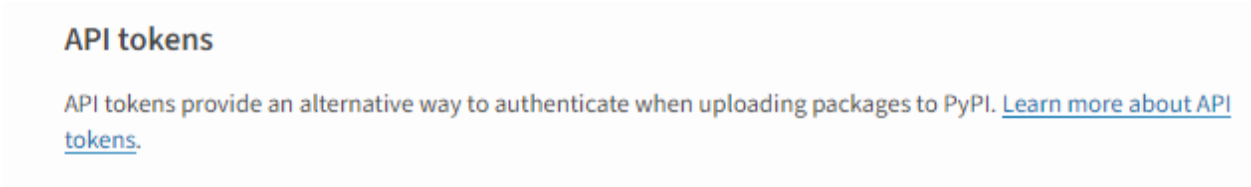
So, I provided step by step guide to create our own python module and upload it to the python pypi website for universal access

Step 1: Create a PyPI Account

- **Go to the PyPI website and create an account.**

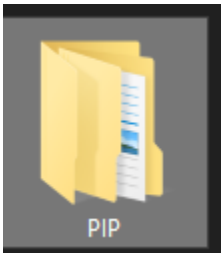
Step 2: Create an API Token

- **Log in to your PyPI account.**
- **Navigate to your account settings and create an API token. This token will be used for uploading your package.**

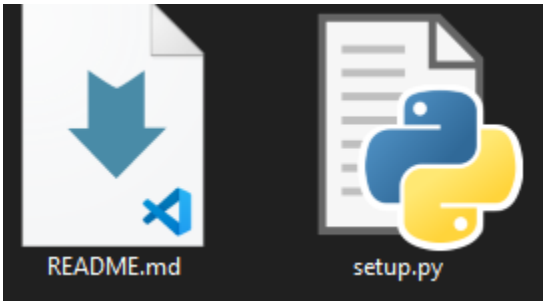


Step 3: Set Up Your Package Directory

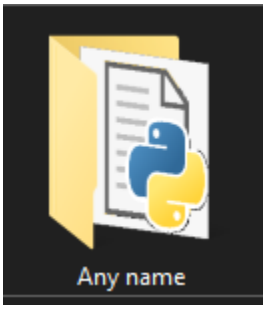
- Create a new directory for your package. Inside this directory, create two files**: README**.md and **setup.py**.
- Also, **create a subdirectory with the same name as your package.** This subdirectory will contain your module files, including an **__init__.py file. Mani folder**



Two files.



Subdirectory Folder.



This is how your directories should look like.

```
your_package/
    README.md
    setup.py
    your_package/
        __init__.py
        module1.py
        module2.py
```

Readme File:

This file contains a comprehensive introduction to your package. It should be stored with a .md extension (Markdown format).

Setup File:

This file contains details about your package, such as the package name, version, description, and other important settings.

Subfolder [module file]:

This folder contains the [init file] and other module files, in case you have many packages. You can create an empty init file, but it should be created because if the init file is not there, your package will not be created.

Step 4: Install Setuptools

Now open setup.py file, and import [setuptools] module.

From that module, import [setup and find packages file].

```
from setuptools import setup, find_packages
```

Step 5: Setup yourSetuptools file

Here you are going to setup your setup file. Ensure that you provide authentic information about yourself and module as well.

This is How your setup file should look like.

```
from setuptools import setup, find_packages
DESCRIPTION = '''
            This is your short description
            '''


setup(
    name='package/module name',
    version='module version',
    description=DESCRIPTION,
```

```
    # Here you are importing your md file
    long_description=open('README.md').read(),
    long_description_content_type='text/markdown',
    author='Your name',
    author_email='<your gmail>',
    url='Your github repository link',
    packages=['The folder name in which the init file
is created before.'],
    classifiers=[
        'Programming Language :: Python :: 3',
        'License :: OSI Approved :: MIT License',
        'Operating System :: OS Independent',
    ],
    python_requires='>=3.6',
)
```

After completing this, now you install an another module called Wheel

```
pip install wheel
```

Step 6: Install Wheel

- **Wheel is a file that stores entire information of module include all files.**
- **In each update of your version, you must create your new wheel that stores the latest version of your module.**
- **This will helps in building your package distribution.**

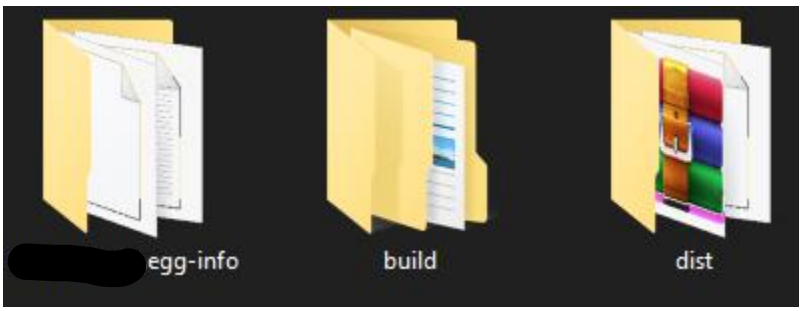
Step 7: Build your paclage

- **Run the following command to create a source distribution and a wheel distribution:**

```
python setup.py sdist bdist_wheel
```

This command will create the following files and directories:

- **dist/: Contains the distribution files (.tar.gz and .whl).**
- **build/: Contains build files.**
- **your_package.egg-info/: Contains metadata about your package.**



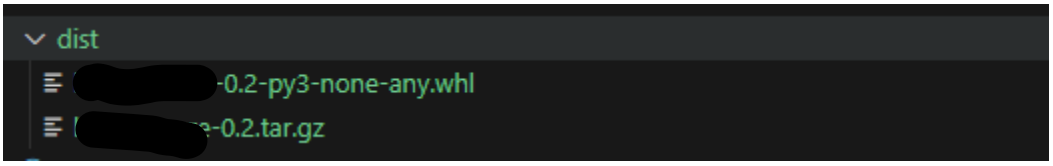
Step 8: Track your Package Locally:

```
cd .\dist
```

- **Navigate to the dist directory and install your package locally to test it. There you run one file for checking your module is working or not.**

```
pip install .\module-name -0.1-py3-none-any.whl
```

you can copy this name from your dist foder:



Copy the name of first file and run the above command

Ensure that you run the command correctly.

This command successfully installs your package into your machine. Then you can easily use your module locally.

For uploading this package into pypi website, where anyone can access your module and use it.

Step 9: Install Twine

```
pip install twine
```

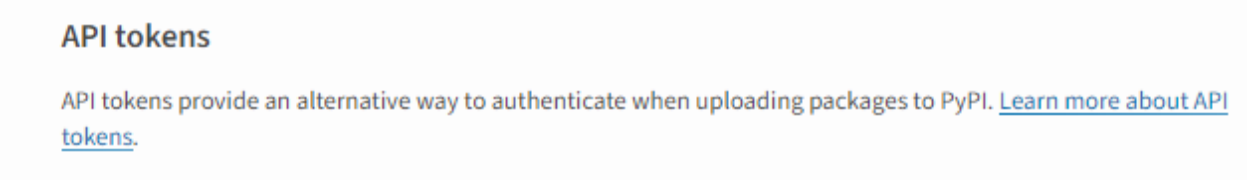
- **This command checks your files. If any error occur you must solved it and again do the same procedure.**

```
twine check dist\*
```

- **This command uploads your package pypi website.**

```
twine upload dist/*
```

- **After Run this command, this will ask you to enter your Api token key. That you were created before.**



Copy that Token and paste it there.

After verification your files are uploaded, and you can easily access your module anywhere in the world.

Congratulations!

Your package is now uploaded to PyPI and can be installed by anyone using:

```
pip install your_package_name
```

Make sure all commands are executed correctly and that your files contain accurate information. Following these steps will help you successfully create and distribute your Python package to the world!