Why I'm doing this?

- **In simply, With just a simple command, 'pip install xxx' we get access to thousands of cool libraries, ready for us to use by doing this.**
- **We can upload upload our own code to PyPi and let others profit from our inventions.**

what is PyPi (pip) ?

- **The Python Package Index (PyPI) is the official third-party software repository for python packages.**
- **PyPI helps us to find and install software developed and shared by the Python community. Whenever we run the pip install command, the pip tool searches the package in this repository and then downloads and installs it to our machine or virtual environment.**

what is a Python Package?

- **Any code that you write in a .py file is known as a module in python. Modules can be imported into another modules.**
- **A collection of more than one modules that target any specific actions can be grouped together to form a package. A package can also contain code organized into directories and sub directories aw well.**

Steps to publish a python package

1. **First, you should remove all "print" statements from your code. But you can use logging.**
2. **Also make sure to not include code that exists outside of a class of a function, otherwise this code will run every time somebody imports your library.**

Create the python package

1. **To create the package, first we should decide the name of our package and then create the directory with the same name of the package. Let us assume we are going to publish a package with the name "novigi_airflow", so the directory should be of the same name. Inside this directory we place all the python files which we need to do packaging. In our scenario we will place all the novigi custom operator files here**

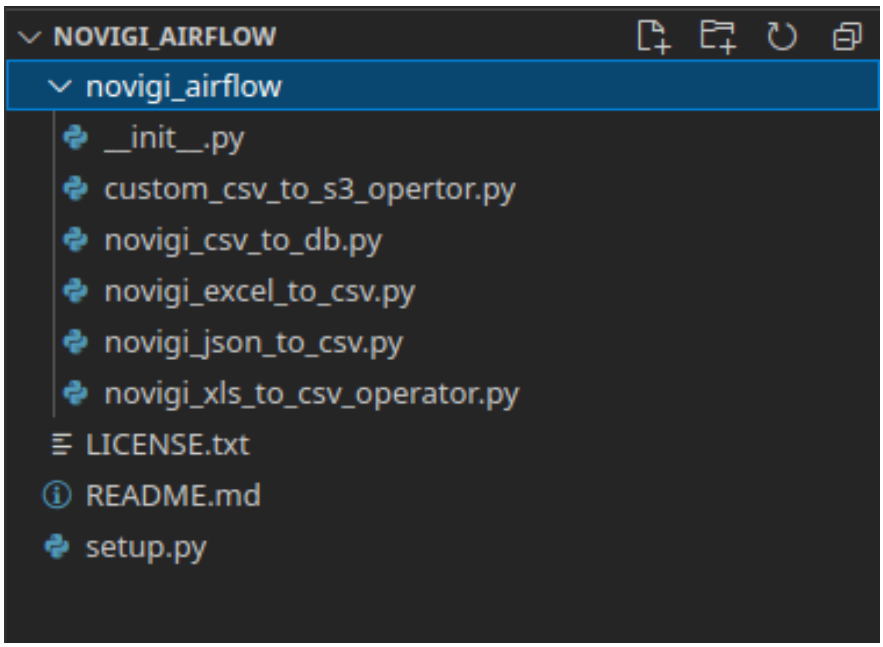
2. **Then Create the following files that PyPi needs, under the root directory.**

    - **setup.py - The setup.py file contains information about our package that PyPi needs, like its name, a description, the current version etc. And also in here we can define all the dependencies our package has(i.e. all the pip packages that we are importing).**
    - **LICENSE.txt - Use this file to define all license details (eg: MIT license).**
    - **README.md - This file is optional but highly recommended**

3. **Now, create a file called __init__.py i (again two underscores) inside the directory that we created with same as the package name. In here the __init__.py** file is used to mark which classes you want the user to access through the package interface. That means **there you and initialize which python classes that users can import from which_package.which_file_name as follows.**

```
from packagename.Filename import Classname
```

In this scenario Our Project structure will be look like follows



Our __init__.py will be as follows

```
from novigi_airflow.custom_csv_to_s3_opertor import CustomCSVToS3Operator
from novigi_airflow.novigi_csv_to_db import NovigiCSVToDatabaseExportOperator
from novigi_airflow.novigi_excel_to_csv import NovigiExcelToCSVExportOperator
from novigi_airflow.novigi_json_to_csv import NovigiJsonToCSVExportOperator
from novigi_airflow.novigi_xls_to_csv_operator import NovigiXlsToCSVOperator
```

Our setup.py will be as follows

```
from setuptools import find_packages, setup

with open("README.md", "r") as fh:
    long_description = fh.read()

setup(
    name='novigi_airflow',
    version="1.0.0",
    description='Novigi Custom Airflow Operators',
    long_description=long_description,
    long_description_content_type='text/markdown',
    license='MIT',
    packages=find_packages(exclude=['tests']),
    install_requires=['requests','jsonpath_ng','pandas'],
    setup_requires=['setuptools', 'wheel'],
    author='Novigi',
    author_email='integration@novigi.com.au',
)
```

Now Create a PyPi account

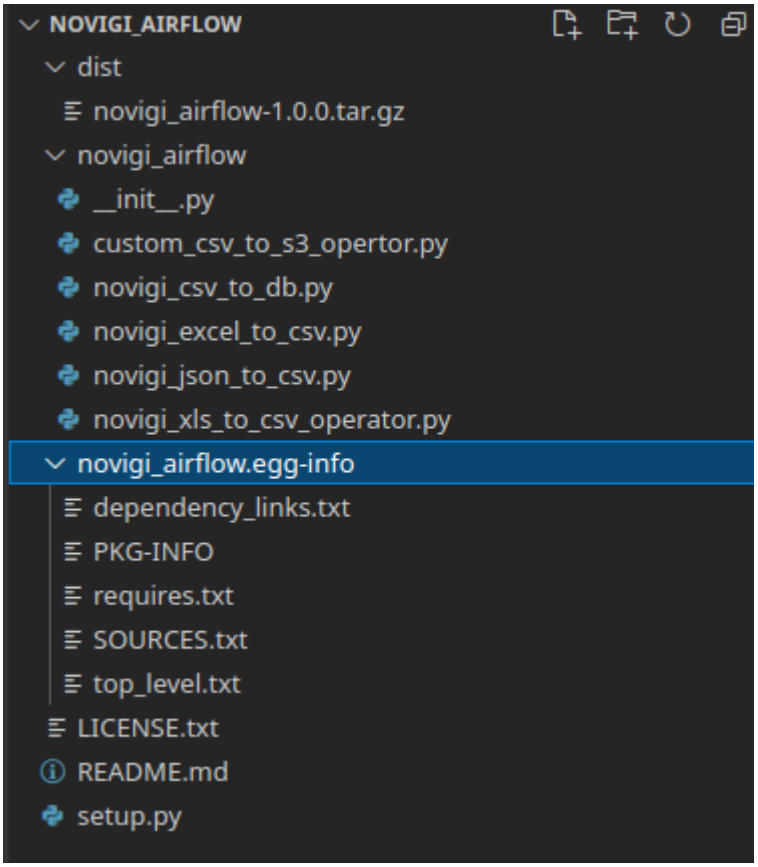
We can register ourselves for a PyPi account and have to remember its username and password, since we will need it later for the upload process.

Ready to Upload our package to PyPi

1. After preparing the package and creating the pipy account, we are ready to distribute it. Before actual distribution, we should make sure that the dependencies required are already existing. To distribute the project, **setuptools** and **wheel** projects are required to be installed. **wheel** project is used to generate **wheel** distribution format. Make sure they are installed and updated by typing the below command in your cmd.

```
pip install --upgrade pip setuptools wheel
```

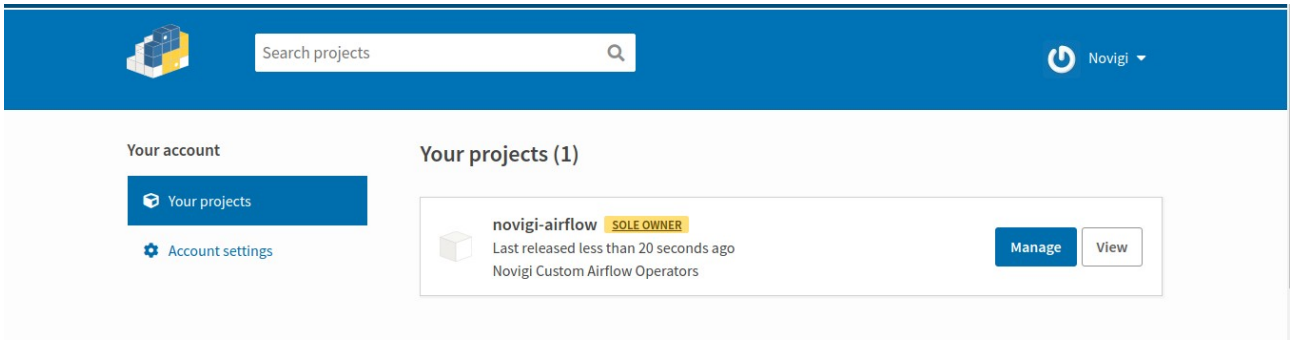
2. The, open the command prompt and navigate into our package folder where we have all our files and our package located. Then create a source distribution with the command: "**python3 setup.py sdist**" . After executing that command our project structure will be look as below with the newly created dist folder.



3. Now We will need **twine** for the upload process, so first install twine via pip: **“pip3 install twine”**

4. Then, run the following command inorder to upload it to our pipy repository: **“twine upload dist/*” (Note that this process will require your PiPy account user_name and password)**

If the uploading process was success, then we can see our package in our Pipy repository as follows.



Now our package is ready to install via pip for anyone in the world by

typing

“sudo pip3 install <package_name>”. In our case “sudo pip3 install novigi_airflow”

Results

```
prefix = None
Collecting novigi_airflow
  Downloading novigi_airflow-1.0.0.tar.gz (5.3 kB)
Requirement already satisfied: requests in ./.local/lib/python3.6/site-packages (from novigi_airflow) (2.25.1)
Requirement already satisfied: jsonpath_ng in ./.local/lib/python3.6/site-packages (from novigi_airflow) (1.5.2)
Requirement already satisfied: pandas in ./.local/lib/python3.6/site-packages (from novigi_airflow) (1.1.5)
Requirement already satisfied: decorator in ./.local/lib/python3.6/site-packages (from jsonpath_ng->novigi_airflow) (5.0.9)
Requirement already satisfied: six in ./.local/lib/python3.6/site-packages (from jsonpath_ng->novigi_airflow) (1.16.0)
Requirement already satisfied: ply in ./.local/lib/python3.6/site-packages (from jsonpath_ng->novigi_airflow) (3.11)
Requirement already satisfied: python-dateutil>=2.7.3 in ./.local/lib/python3.6/site-packages (from pandas->novigi_airflow) (2.8.1)
Requirement already satisfied: numpy>=1.15.4 in ./.local/lib/python3.6/site-packages (from pandas->novigi_airflow) (1.19.5)
Requirement already satisfied: pytz>=2017.2 in ./.local/lib/python3.6/site-packages (from pandas->novigi_airflow) (2021.1)
Requirement already satisfied: idna<3,>=2.5 in ./.local/lib/python3.6/site-packages (from requests->novigi_airflow) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in ./.local/lib/python3.6/site-packages (from requests->novigi_airflow) (1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in ./.local/lib/python3.6/site-packages (from requests->novigi_airflow) (2020.12.5)
Requirement already satisfied: chardet<5,>=3.0.2 in ./.local/lib/python3.6/site-packages (from requests->novigi_airflow) (4.0.0)
Building wheels for collected packages: novigi-airflow
  Building wheel for novigi-airflow (setup.py) ... done
  Created wheel for novigi-airflow: filename=novigi_airflow-1.0.0-py3-none-any.whl size=8009 sha256=f1a8626bbc224f83f771902b5da0aefd64024b9e80151bab3d2d245e9e62c5db
  Stored in directory: /tmp/pip-ephem-wheel-cache-p218uxj8/wheels/46/c7/f3/c1b4cbf6f5d3d670fba8fe5e85cec9df5bf126b22ab7776eaa
Successfully built novigi-airflow
Installing collected packages: novigi-airflow
  WARNING: Value for scheme.headers does not match. Please report this to <https://github.com/pypa/pip/issues/10151>
  distutils: /usr/local/include/python3.6/novigi-airflow
  sysconfig: /usr/include/python3.6m/novigi-airflow
Successfully installed novigi-airflow-1.0.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual en
vironment instead: https://pip.pypa.io/warnings/venv
prasadi@prasadi-HP-ProBook-450-G5 ~ $
```

When Changing our package

If we maintain our package well, we will need to change the source code form time to time. Follow the bellow steps if you did any source code change.

1. Made the relevant changes in files inside the package folder.

2. Then adapt the setup.py file (according to our new release tag, new version etc.), then go inside the package folder and run the setup.py and the twin command again.

```
1) python3 setup.py sdist
2) twine upload dist/*
```

Credentials for PiPy account

```
User_Name : Novigi
Password : NovigiLK@123
```

3. Finally, update our package via pip to see whether our changes worked:

```
pip install novigi_airflow --upgrade
```

When Deploying in our environments

Some thing to keep remember when we do pip install is that, do it as a supper user in order to install package in the correct python path. That means always type

sudo pip install package_name instead of just typing pip install package_name.

In our case type

```
sudo pip install novigi_airflow
```

If you type just pip install package_name as a normal user then your packages will be installed to your .local directory which makes error when trying to import those installed packages in our dags , which says "No module Found"

Relevant Link : https://unix.stackexchange.com/questions/240037/why-did-pip-install-a-package-into-local-bin

In case if installing the package as a supper user and if its still gives a "No module Found" error what you can do is go to the correct python directory that package should be installed and type

```
sudo pip install package_name -t .
```

in our case it will be as follows

```
sudo pip install novigi_airflow -t .
```

Now you can import these operators and other things in these packages inside our airflow dags as follows.

```
from packagename.Filename import Classname
```

Related to our pakage it will be as follows

```
from novigi_airflow.custom_csv_to_s3_opertor import CustomCSVToS3Operator

from novigi_airflow.novigi_csv_to_db import NovigiCSVToDatabaseExportOperator

from novigi_airflow.novigi_excel_to_csv import NovigiExcelToCSVExportOperator

from novigi_airflow.novigi_json_to_csv import NovigiJsonToCSVExportOperator
```

By importing them as that in above we can make use of these operators inside our dags as the same way that we importing them from direct airflow operator folder .

Credentials for PiPy account

```
User_Name : Novigi

Password : NovigiLK@123
```

Git Repo Link : https://bitbucket.org/novigi/novigi_airflow/src/master/

Useful References:

- https://www.kdnuggets.com/2018/06/packaging-distributing-python-project-pypi-pip.html/2
- https://airflow.apache.org/docs/apache-airflow/stable/modules_management.html

__init__.py file

- If you specify your package name specially as below in setup.py file , then you do not need to include __init__.py file in any directly(root or sub directories)

```
packages=['airflow_customs_by_novigi','airflow_customs_by_novigi.operators',
'airflow_customs_by_novigi.hooks', 'airflow_customs_by_novigi.custom_plugings'],
```

But it will not gives an error though you specify the init file. If you are writing any content on it, write it as follows

```
from airflow_customs_by_novigi.operators.custom_csv_to_s3_opertor import
CustomCSVToS3Operator
```

- **But if you are writing that as below in setup.py file , then you have to initialize a __init__.py in root directory (just inside the package folder) and in each sub directory locations in it.**

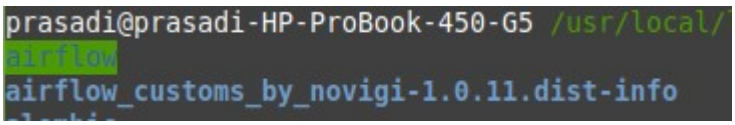
```
from setuptools import setup, find_packages

setup(name='MySoftware',
      packages=find_packages()
  )
```

- **when you write setup.py file as above then you must have __init__.py file in each directory. But you can keep it blank.. You do not need to specify the content. If you are specify any content specify it correctly as follows**

```
from airflow_customs_by_novigi.operators.custom_csv_to_s3_opertor import CustomCSVToS3Operator
```

- **If you do not include __init__.py files in this senario you want get an error when making and uploading the package. But it will store in the /usr/local/lib/python3.6/dist-packages/ directry as follows.**



But you can't use these package contents inside an airflow dags. It wil gives "No module found error"

- **When you do the work correctly by including the __init__.py file, then our package will store in the /usr/local/lib/python3.6/dist-packages/ directry as follows in the correct manner.**