Distributed custom packages

Purpose of distributing custom packages

Distributing a custom package in Python typically involves creating a Python package,

building a distribution package (e.g., a source distribution or a wheel),

And then sharing it through a package distribution system

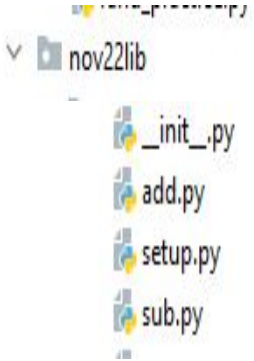
The most common tool for this task is setuptools along with for creating wheel binary distributions.

In other words we can say that this process is used to distribute our user defined package to others

Steps involved

1. Organize Your Project:

Ensure that your project is structured as a Python package. This includes having a proper directory structure with a setup.py file and package directory with your module files



1. Create setup.py:

Create a setup.py file att of your project. This file contains metadata about your package, such as the package name, version, description, author, etc.

```
from setuptools import setup, find_packages

setup(
    name='nov22lib',
    version='1.0',
    Summary = 'Easily download, build, install, upgrade, and uninstall Python packages',
    Author='Hemalatha',
    Author_email='hemalatha.chandaka@thundersoft.com' ,
    Location= 'C:/Users/dell/AppData/Local/Programs/Python/Python311/Lib/site-packages',
    packages=find_packages(),

install_requires=[
        # list your dependencies here

    ],
)
```

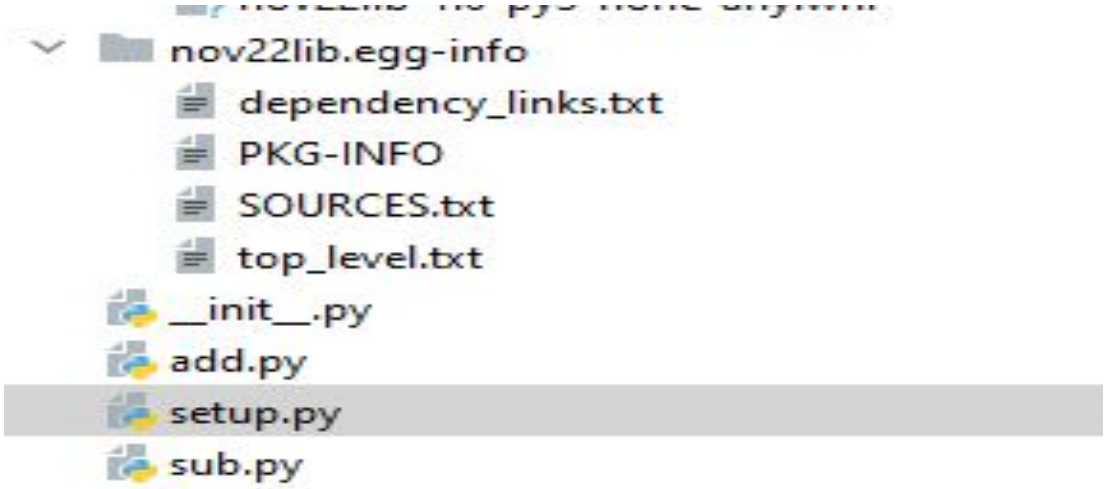
3. Build the distribution package:

Open a terminal in the directory containing setup.py and run the following command to build the distribution package:

```
python setup.py bdist_wheel
```

This command creates a source distribution sdist and a wheel distribution bdist wheel The wheel format is a binary distribution format and is preferred for distribution.

→ Now our package will be looks like this



4. Install twine if not installed:

```
pip install twine
```

5. Create pypi account:

If you don't have a PyPI account, you need to create one.

Visit https://pypi.org/account/register/

6. Upload to pypi:

Go to terminal and give the command → twine upload dist/*

Once it gets successfully uploaded we got below lines in terminal

```
Uploading nov22lib-1.0-py3-none-any.whl
100% ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 kB • 00:00 • ?
Uploading nov22lib-1.0.tar.gz
100% ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 kB • 00:00 • ?

View at:
https://pypi.org/project/nov22lib/1.0/
PS C:\Users\dell\PycharmProjects\pythonProject4\nov22lib>
```

→if we got any issue while uploading then do below steps

1. Generate a PyPI API Token:

Log in to your PyPI account on https://pypi.org/manage/account/token/.
Click on "Add API Token."
Provide a description for the token (e.g., "Twine Upload").
Click on "Generate Token."

Copy the generated API token.

2. Update Your `~/.pypirc` File:

Update your ~/.pypirc file in your home directory to use the generated API token instead of your password. Here is an example:

→ now try to upload again by the below command

```
twine upload dist/*
```

7. Install from pypi:

pip install package-name

```
PS C:\Users\dell\PycharmProjects\pythonProject4> pip install nov22lib
Collecting nov22lib
  Downloading nov22lib-1.0-py3-none-any.whl.metadata (55 bytes)
Downloading nov22lib-1.0-py3-none-any.whl (922 bytes)
Installing collected packages: nov22lib
Successfully installed nov22lib-1.0
PS C:\Users\dell\PycharmProjects\pythonProject4>
```

8.Share the pypi

pip install your-package-name

Install the package where required

9. Verify package installation:

pip list

To check whether custom package is installed or not

→finally work with that package by importing it

Difference between bdist_wheel and bdist_egg

Bdist_wheel and bdist_egg are both commands used for building binary distribution packages in Python, but they are associated with different packaging formats. Here's a breakdown of the key differences between them:

1. Package format

Wheel (`*.whl`)

Egg (`*.egg`)

2. Advantages

```
python setup.py bdist_wheel
```

```
python setup.py bdist_egg
```

3. Advantages

bdist_wheel:

Faster installation compared to source distributions.

Supports direct installation of binary packages without needing to build.

bdist_egg:

Introduced features like namespace packages and entry points.

Bundles metadata and Python code into a single file.

4.Dependency

Requires the `wheel` package to be installed (`pip install wheel`).

Part of the standard library and Doesn’t require additional package.

5.Metadata

The binary distribution is in a Specific directory structure

eggs contain metadata in a directory name EGG-INFO