



Library Creation in PyPI

PyPI stands for **Python Packaging Index** is the official software **repository** for python. All libraries can be easily accessed with "**pip install <package>**" command. Over **400,000+** Python Packages are available and there is **no restriction** on the usability or the intention of the codes.

This blog will outline the library creation in **5 simple steps**. For this purpose, we will create a **Caesar Cipher Encryption and Decryption** library called **cipherLV** in PyPI.

cipherLV is available in PyPI at  [cipherLV](#)

cipherLV is available in github at  [GitHub - sarathbabu-latentview/lv_cipher](#)

"PyPI is what makes Python Powerful"

Advantages

- **Independent: PyPI is platform independent**
 - can be used across different operating systems such as Windows, MacOS , Linux etc..
- **Migration:** All custom libraries can be **migrated** across **different clients**.
 - Once uploaded to PyPI, codes can be easily downloaded and can be used across different projects.
 - It also allows the project members to share and access code thereby allowing inter-operability
- **Ease of Development:** PyPI libraries are **easy to upload** to PyPI repository
- **Recognition:** Individual **recognition** based on package uploads

Requirements

- Python 3 version
- Python IDE
- Command Prompt
- PyPI Account
- GitHub Account

STEP 1: Create the Directory Structure

- Create a new repository with your library name in **GitHub** along with **README.MD** and **LICENSE**
- Download the repository to your local drive
- The directory should be of the following structure:

Folder Name

cipherLV

- Library Folder Name
 - __init__.py
 - Actual Code.py
- setup.py
- README.MD
- LICENSE

- cipherLV
 - __init__.py
 - cipherLV.py
- setup.py
- README.MD
- LICENSE

STEP 2: Code Format

- The code should contain **Modules** and **Classes**
- The code should contain **DOCSTRING** after Module Declaration

```

1  def encrypt(text,s=3):
2  """Performs Encryption of the text data with a symmetric key number
   s.
3      Parameters:
4          text(string) : Input text to be encrypted
5          s(int) : Default value = 3, Symmetric Key number to perform
   ciphering
6      Returns:
7          string: Encrypted Input text
8  """
9      result = ""
10     for i in range(len(text)):
11         char = text[i]
12         result += chr((ord(char) + s - 97) % 26 + 97)
13     return result

```

Declaration

DOCSTRING

Actual Code

STEP 3: Setup File Format

setup.py is used to **package** the library to be uploaded to PyPI Site. Setup File should contain the following.

1. Name
2. Author Name
3. Author Email Address
4. Description
5. Long Description (Read from README.MD to be displayed in PyPI site)
6. Long Description Content Type (Plain Text, Markdown File)
7. Code Location
8. Dependency Requirements (other packages to be installed from PyPI)
9. Keywords to be searched for in PyPI
10. Classifier (Project Phase, Intended Audience, Operating Systems)

```

1  setup(
2      name="cipherLV",
3      version='0.1.4',
4      author="sarath babu",
5      author_email="sarathbabu.karunanithi@latentview.com",
6      description='Description: LV cipher is used for encrypting and decrypting with symmetric key',
7      long_description_content_type="text/markdown",
8      long_description=long_description,

```

```

9 packages=find_packages()
10 py_modules=['cipherLV'],
11 install_requires=['sklearn','pandas'],
12 keywords=['python', 'encryption','decryption','cryptography'],
13 classifiers=[
14     "Development Status :: 1 - Planning", "Intended Audience :: Developers",
15     "Programming Language :: Python :: 3",
16     "Operating System :: Unix", "Operating System :: MacOS :: MacOS X",
17     "Operating System :: Microsoft :: Windows"])
18

```

STEP 4: README File Format

README.MD is used to **describe** the library in detail in PyPI website. Being a markdown file, README contains description in **HTML format**. The contents of README are displayed in PyPI website.

```

1 <h1> cipher LV</h1>
2 <h2> cipher LV is used for encrypting and decrypting with symmetric key</h2>
3 <p> cipher LV has two modules </p>
4 <ul>
5 <li>Encrypt</li>
6 <li>Decrypt</li>
7 </ul>

```

STEP 5: Package and Upload

Setup.py is used to **package** the library which, is done with the following commands in command prompt

```

1 pip install setuptools
2 python setup.py sdist bdist_wheel

```

Packaged library is uploaded to PyPI with the following commands

```

1 pip install twine
2 twine upload dist/*

```

twine upload will prompt the user for PyPI username and password.

Demo

```

(base) C:\Users\sarathbabu.karunanithi\Downloads\Tutorials\PyPI Tutorials\cipherLV>python setup.py sdist bdist_wheel
running sdist
running egg_info
creating cipherLV.egg-info
writing cipherLV.egg-info\PKG-INFO
writing dependency_links to cipherLV.egg-info\dependency_links.txt
writing top-level names to cipherLV.egg-info\top_level.txt
writing manifest file 'cipherLV.egg-info\SOURCES.txt'

```

```

(base) C:\Users\sarathbabu.karunanithi\Downloads\Tutorials\PyPI Tutorials\cipherLV>twine upload dist\*
Uploading distributions to https://upload.pypi.org/legacy/
Enter your username: sarathbabu.karunanithi
Enter your password:
Uploading cipherLV-0.1.4-py3-none-any.whl
100% ----- 6.7/6.7 kB • 00:01 • ?
Uploading cipherLV-0.1.4.tar.gz
100% ----- 5.7/5.7 kB • 00:00 • ?

View at:
https://pypi.org/project/cipherLV/0.1.4/


```

cipherLV 0.1.4

✓

Latest version

`pip install cipherLV`



Released: May 15, 2023

Description: LV cipher is used for encrypting and decrypting with symmetric key

Navigation

Project description

Release history

Download files

Statistics

View statistics for this project via [Libraries.io](#), or by using [our public](#)

Project description

cipher LV

cipher LV is used for encrypting and decrypting with symmetric key

cipher LV has two modules

- Encrypt
- Decrypt