# Python实验18 项目结构

## 一、实验介绍

本实验阐述了一个完整的 Python 项目结构，你可以使用什么样的目录布局以及怎样发布软件到网络上。

知识点

创建项目，编写 \_\_init\_\_ 文件

使用 setuptools 模块，编写 setup.py 和 MANIFEST.in 文件

创建源文件的发布版本

项目注册&上传到 PyPI

## 二、创建 Python 项目

我们的实验项目名为 factorial，放到 /home/shiyanlou/factorial 目录：

$ cd /home/shiyanlou

$ mkdir factorial

$ cd factorial/

我们给将要创建的 Python 模块取名为 myfact，因此我们下一步创建 myfact 目录。

$ mkdir myfact

$ cd myfact/

主代码将在 fact.py 文件里面。

"myfact module"

def factorial(num):

"""

返回给定数字的阶乘值

:arg num: 我们将计算其阶乘的整数值

:return: 阶乘值，若传递的参数为负数，则为 -1

"""

if num >= 0:

if num == 0:

return 1

return num \* factorial(num -1)

else:

return -1

我们还有模块的 \_\_init\_\_.py 文件，内容如下：

from fact import factorial

\_\_all\_\_ = [factorial, ]

我们还在 factorial 目录下添加了一个 README.rst 文件。因此，目录结构看起来像下面这样：

2.1 MANIFEST.in

现在我们要写一个 /home/shiyanlou/factorial/MANIFEST.in 文件，它用来在使用 sdist 命令的时候找出将成为项目源代码压缩包一部分的所有文件。

include \*.py

include README.rst

如果你想要排除某些文件，你可以在这个文件中使用 exclude 语句。

2.2 安装 python-setuptools 包

我们使用 virtualenv（这里不示范步骤）。

$ sudo pip3 install setuptools

2.3 setup.py

最终我们需要写一个 /home/shiyanlou/factorial/setup.py，用来创建源代码压缩包或安装软件。

#!/usr/bin/env python3

"""Factorial project"""

from setuptools import find\_packages, setup

setup(name = 'factorial',

version = '0.1',

description = "Factorial module.",

long\_description = "A test module for our book.",

platforms = ["Linux"],

author="ShiYanLou",

author\_email="support@shiyanlou.com",

url="https://www.shiyanlou.com/courses/596",

license = "MIT",

packages=find\_packages()

)

name 是项目名称，version 是发布版本，description 和 long\_description 分别是项目介绍，项目长描述。platforms 是此模块的支持平台列表。find\_packages() 是一个能在你源目录下找到所有模块的特殊函数，packaging docs。

2.3.1. setup.py 用例

要创建一个源文件发布版本，执行以下命令。

$ python3 setup.py sdist

执行完毕会返回类似下面的信息：

running sdist

running egg\_info

creating factorial.egg-info

writing factorial.egg-info/PKG-INFO

writing top-level names to factorial.egg-info/top\_level.txt

writing dependency\_links to factorial.egg-info/dependency\_links.txt

writing manifest file 'factorial.egg-info/SOURCES.txt'

reading manifest file 'factorial.egg-info/SOURCES.txt'

reading manifest template 'MANIFEST.in'

writing manifest file 'factorial.egg-info/SOURCES.txt'

running check

creating factorial-0.1

creating factorial-0.1/factorial.egg-info

creating factorial-0.1/myfact

making hard links in factorial-0.1...

hard linking MANIFEST.in -> factorial-0.1

hard linking README.rst -> factorial-0.1

hard linking setup.py -> factorial-0.1

hard linking factorial.egg-info/PKG-INFO -> factorial-0.1/factorial.egg-info

hard linking factorial.egg-info/SOURCES.txt -> factorial-0.1/factorial.egg-info

hard linking factorial.egg-info/dependency\_links.txt -> factorial-0.1/factorial.egg-info

hard linking factorial.egg-info/top\_level.txt -> factorial-0.1/factorial.egg-info

hard linking myfact/\_\_init\_\_.py -> factorial-0.1/myfact

hard linking myfact/fact.py -> factorial-0.1/myfact

Writing factorial-0.1/setup.cfg

creating dist

Creating tar archive

removing 'factorial-0.1' (and everything under it)

我们能在 dist 目录下看到一个 tar 压缩包。

$ ls dist/

factorial-0.1.tar.gz

记住尝试安装代码时使用 virtualenv。

执行下面的命令从源代码安装。

$ sudo python3 setup.py install

学习更多可前往 packaging.python.org。

2.4 Python Package Index (PyPI)

你还记得我们经常使用的 pip 命令吗？有没有想过这些包是从哪里来的？答案是 PyPI。这是 Python 的软件包管理系统。

为了实验，我们会使用 PyPI 的测试服务器 https://testpypi.python.org/pypi。

2.4.1 创建账号

首先在这个链接注册账号。你会收到带有链接的邮件，点击这个链接确认你的注册。

创建 ~/.pypirc 文件，存放你的账号详细信息，其内容格式如下：

[distutils]

index-servers = pypi

testpypi

[pypi]

repository: https://upload.pypi.org/legacy/

username: <username>

password: <password>

[testpypi]

repository:https://test.pypi.org/legacy/

username: <username>

password: <password>

替换 <username> 和 <password> 为您新创建的帐户的详细信息。在这里，由于我们是到 testpypi的网页上去注册账号，即将相应的服务上传到 testpypi，所以在这里，你只需修改[testpypi]的用户名和密码

记得在 setup.py 中更改项目的名称为其它的名字来测试下面的指令，在接下来的命令中我将项目名称修改为factorial2，为了不重复，需要自行修改至其它名称。

2.4.2 上传到 TestPyPI 服务

下一步我们会将我们的项目到 TestPyPI 服务。这通过 twine 命令完成。

我们也会使用 -r 把它指向测试服务器。

$ sudo pip3 install twine

$ twine upload dist/\* -r testpypi

执行完毕会返回类似下面的信息,

Uploading distributions to https://test.pypi.org/legacy/

Uploading factorial2-0.1.tar.gz

在这里你也可以使用下面的命令上传到 PyPI 服务上，但这里需要注意，在 ~/.pypirc 里面，你需要到 https://pypi.python.org页面，按照上面的步骤去注册一个账号，然后到~/.pypirc 的 [pypi] 下填写相应的用户名和密码。testpypi 和 pypi 的账号密码并不通用。

$ twine upload dist/\* -r pypi

现在如果你浏览这个页面，你会发现你的项目已经准备好被别人使用了。

## 三、总结

实验知识点回顾：

创建项目，编写 \_\_init\_\_ 文件

使用 setuptools 模块，编写 setup.py 和 MANIFEST.in 文件

创建源文件的发布版本

项目注册&上传到 PyPI

本实验使用了 setuptools 包，并完成了较为完整的项目创建&发布流程，最后还将项目发布到了网络 （PyPI）上。