

How to package your python code to library

井民全, Jing, mqjing@gmail.com

為了避免 import error 的惡夢, 避免在程式裡寫一堆參考來參考去的相對import路徑, 讓自動測試 ready 的機制, 以及提供自己的 module 將來可以透過 <https://travis-ci.org/> 進行自動整合測試. 我們打算在一開始就考慮 適當的 code layout, package module import 以及 pytest auto-test.

這份文件包含了

1. Python Code: 好的 code layout 長什麼樣 ([go](#)) ([好處解釋 1](#), [好處解釋 2](#))
2. Install Package: 如何撰寫 setup.py, 安裝你的 python code ([go](#))
 - a. Develop mode: 直接編修程式碼, 不用擔心還要再安裝
 - b. User mode: 搞定後, 模擬使用者直接安裝 model
3. 最簡單的 pytest 範例, 使用 qbot 模擬滑鼠箭排操作 UI ([go](#))
 - a. 安裝 pytest
 - b. 範例程式

1. Python Code: 好的 code layout 長什麼樣

1.1. Setup environment

```
conda create --name venv_demo python=3 # 要求使用 python 3 最新版, 當 venv
conda activate venv_demo
conda install pip # 使用 conda 安裝 pip, pip 會根據你的 venv 決定 pip2 或 pip3. 目前是 pip3
```

1.2. Code structure

```
root/
  setup.py      # 安裝模組的 script (透過 setup.py, 可以只安裝 src, 不安裝 test module)
  src/          # <---- 你的程式碼放在 src 目錄裡面
    mypkg/      # package 模組 1: mypkg
      __init__.py
```

```
app.py
view.py

mypkg2/          # package 模組 2: mypkg2
__init__.py
module1.py
module2.py

tests/           # <---- 所有測試的程式碼, 放在 tests 裡面
__init__.py
test_pkg1/
__init__.py
test_view.py
test_pkg2/
__init__.py
test_module.py
```

1.3. Code

```
# File: src/mypkg/view.py
from PyQt5.QtWidgets import QMainWindow, QApplication, QWidget, QAction
class View(QWidget):
    def __init__(self):
        print('this is View')
```

2. Install package

Setup.py

```
from setuptools import setup, find_packages
setup(
    name="mypackage2",
    packages=find_packages('src'),    # 指定安裝放在 src 底下的 packages
    package_dir={'': 'src'}    # package root 從目前目錄 換成 ./src 目錄下 (ref)
)

# 因為 find_packages 與 package_dir 設定的關係, test module 下的所有 code 都不會安裝進系統
```

Option 1: Develop mode:

直接在 src 目錄修改程式, 不用每次修改 code 都要再安裝一次

Command

```
pip install -e .
```

Verify

check the 'mypackage2'

```
pip list
```

Option 2: User mode

如同使用者收到這包 package, 安裝

Command

```
pip install .
```

Verify

check the 'mypackage2'

```
pip list
```

check where mypackage2 installed

```
pip show mypackage2
```

check package

你在 src 的 package mypkg 會安裝在

~/anaconda3/envs/venv_pytest/lib/python3.8/site-packages 裡 mypkg 的目錄裡

當發現 import 找不到時, 直接到

~/anaconda3/envs/venv_pytest/lib/python3.8/site-packages/mypkg 裡面去找. source code 全在這

2.1. Use the package

在任何地方都可以使用

```
from mypkg.view import View
```

3. Auto test

Step 1: Install pytest

Command

```
pip install pytest-qt
```

Step 2: Write the test code

File: root/tests/test/test_pkg1/test_view.py

```

import sys
import pytestqt.qtbot as qtbot
from mypkg.view import View

class t_ViewEditWidget_App(QMainWindow):
    def __init__(self):
        super().__init__()

    def __init_UI__(self):
        self.widget_view = View(None)
        self.setCentralWidget(self.widget_view)
        self.show()

def test_gui_auto(qtbot):
    app = QApplication(sys.argv)
    ex = t_ViewEditWidget_App()
    qtbot.add_widget(ex)
    ex.__init_UI__()

    import time
    qtbot.keyClicks(ex.widget_view.comb_edit, '2')
    #time.sleep(2)
    qtbot.keyClicks(ex.widget_view.comb_edit, '1')
    print('ok')
    app.exec_()

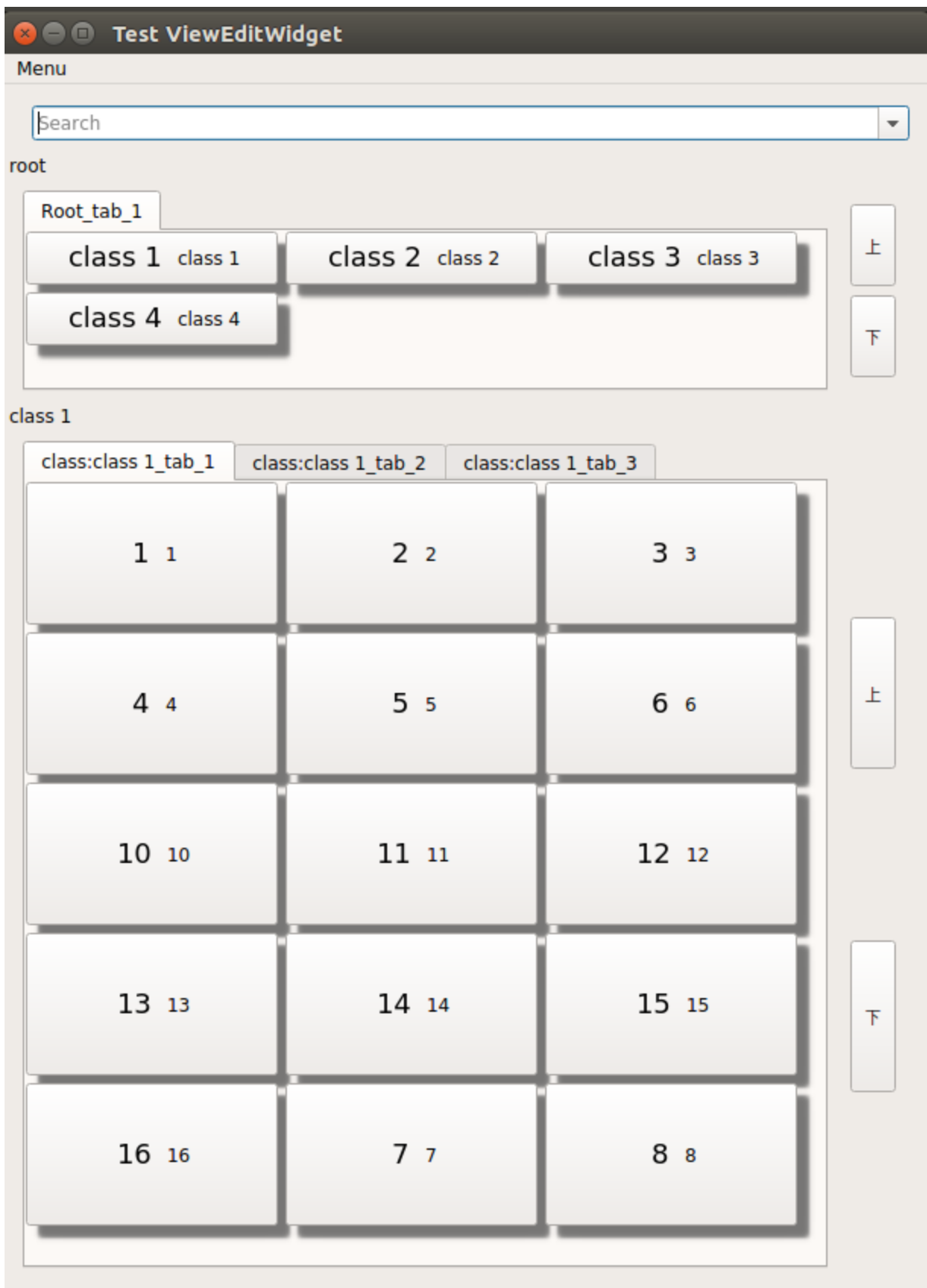
```

Step 3: Run the test

cd tests/test_pkg1/

python3 -m pytest test_view.py

示意圖



4. Uninstall package

pip uninstall `package-name`

Ex:

pip uninstall `mypackage2`

5. References

1. <https://blog.ionelmc.ro/2014/05/25/python-packaging/#the-structure>
2. <https://docs.pytest.org/en/latest/goodpractices.html#test-discovery>