Прикључци у Пајтону

Prof. dr Igor Dejanović (igord at uns ac rs)

Kreirano 2024-09-30 Mon 14:25, pritisni ESC za mapu, m za meni, Ctrl+Shift+F za pretragu

Sadržaj

Мотивација

- Пакет жели да омогући команду која се може позвати са терминала.
- Пакет жели да омогући покретање GUI апликације.
- Пакет жели да омогући прилагођавање своје функционалности употребом прикључака (*plugins*).

Конзолне скрипте

Организација пројекта

```
project_root_directory
├── pyproject.toml        # and/or setup.cfg, setup.py
└── src
    └── timmins
        ├── __init__.py
        └── ...
```

у `__init__.py`:

```python
def hello_world():
    print("Hello world")
```

Покретање као модул

у `src/timmins/__main__.py`:

```
from . import hello_world

if __name__ == '__main__':
    hello_world()
```

Сада је могуће покренути функцију са терминала на следећи начин:

```
$ python -m timmins
Hello world
```

Преко екстензионе тачке

У `pyproject.toml`:

```
[project.scripts]
hello-world = "timmins:hello_world"
```

Сада можемо скрипту директно позвати:

```
$ hello-world
Hello world
```

Исто тако можемо регистровати и GUI скрипту

```
[project.gui-scripts]
hello-world = "timmins:hello_world"
```

Генеричко решење

- Регистрација конзолних и GUI скрипти представља специјалну употребу општег механизма за проширење који називамо *Прикључне тачке*.

Прикључне тачке (*Entry Points*)

- Општи начин проширења функционалности пакета.
- Тип метаподатка који пакети могу да региструју.
- Други пакети могу динамички да открију и исчитају ове метаподатке.

Организација пројекта

Крећемо од исте организације пројекта.

```
project_root_directory
├── pyproject.toml         # and/or setup.cfg, setup.py
└── src
    └── timmins
        ├── __init__.py
        └── ...
```

у `__init__.py`:

```python
def hello_world():
    print("Hello world")
```

Поставка проблема

- Желимо да омогућимо прилагођен приказ основног стринга.
- Прилагођавање могу да ураде други пакети.

  На пример испис може бити промењен на следећи начин:

```
!!! Hello world !!!
```

Први корак

Раздвајање приказа на две функције:

```
def display(text):
    print(text)

def hello_world():
    display('Hello world')
```

Откривање прикључака и модификоване `display` функције

```
from importlib.metadata import entry_points
display_eps = entry_points(group='timmins.display')
```

Прикључне тачке имају назив, групу и вредност (атрибути `name`, `group`, `value`):

```
(
    EntryPoint(name='excl', value='timmins_plugin_fancy:excl_display', group='timmins.display'),
    ...,
)
```

Учитавање објекта регистрованог у прикључној тачки

```
display = display_eps[0].load()
```

Потребно је дефинисати понашање у случају да не постоји пакет који региструје функцију у прикљуној тачки:

```
from importlib.metadata import entry_points
display_eps = entry_points(group='timmins.display')
try:
    display = display_eps[0].load()
except IndexError:
    def display(text):
        print(text)

def hello_world():
    display('Hello world')
```

Регистрација прилагођене `display` функције

```
timmins-plugin-fancy
├── pyproject.toml        # and/or setup.cfg, setup.py
└── src
    └── timmins_plugin_fancy
        └── __init__.py
```

у `src/timmins_plugin_fancy/__init__.py`:

```python
def excl_display(text):
    print('!!!', text, '!!!')
```

`pyproject.toml`

```toml
# Note the quotes around timmins.display in order to escape the dot .
[project.entry-points."timmins.display"]
excl = "timmins_plugin_fancy:excl_display"
```

Сада можемо видети дејство прикључка уколико је пакет инсталиран:

```
>>> from timmins import hello_world

>>> hello_world()
!!! Hello world !!!
```

У случају да пакет `timmins_plugin_fancy` није инсталиран понашање је подразумевано:

```
>>> from timmins import hello_world

>>> hello_world()
Hello world
```

Регистрација више прикључака под истом групом

- Желимо да имамо више различитих начина исписа:

```python
def excl_display(text):
    print('!!!', text, '!!!')

def lined_display(text):
    print(''.join(['-' for _ in text]))
    print(text)
    print(''.join(['-' for _ in text]))
```

`pyproject.toml`

```toml
[project.entry-points."timmins.display"]
excl = "timmins_plugin_fancy:excl_display"
lined = "timmins_plugin_fancy:lined_display"
```

Сада можемо у базном пакету урадити нешто попут овога:

```
display_eps = entry_points(group='timmins.display')
try:
    display = display_eps['lined'].load()
except KeyError:
    # if the 'lined' display is not available, use something else
    ...
```

Или можемо учитати све прикључке:

```
display_eps = entry_points(group='timmins.display')
for ep in display_eps:
    display = ep.load()
    # do something with display
    ...
```

Литература

- Setuptools Entry Points
- Creating and discovering plugins