
syringe

Release 0.4.0

Mar 02, 2018

Contents

1	Usage	3
1.1	Mocking	4
	Python Module Index	5

A simple dependency injection library.

CHAPTER 1

Usage

First decorate a class with `@provides('a lookup name')`.

```
>>> import syringe
>>>
>>> @syringe.provides('cure')
... class Syrup:
...     def drink(self, person):
...         print('Nom nom')
...         person.health = 100
... 
```

Instantiate it. Note that it is not possible to instantiate another instance of a class decorated with the name `cure`.

```
>>> syrup = Syrup()
```

Next inject it in another class using `inject('a lookup name')`.

```
>>> class Person:
...     cure = syringe.inject('cure')
...
...     def drink_medicine(self):
...         self.cure.drink(self)
... 
```

When an instance of the `Person` class is created, the value of the injecte name is the instance of the provided and instantiated class.

```
>>> person = Person()
>>> person.health = 20
>>> assert person.cure == syrup
>>> person.drink_medicine()
Nom nom
>>> assert person.health == 100
```

1.1 Mocking

A mock instance can be inserted using `syringe.mock('a lookup name')`

```
>>> try:
...     from unittest import mock
... except:
...     import mock
...
>>> m = syringe.mock('cure')
>>> person.drink_medicine()
>>> m.drink.assert_called_once_with(person)
```

`syringe.clear()`

Clears all current providers.

This may be useful for testing.

exception `syringe.DuplicateProviderError`

Bases: `exceptions.ValueError`

Raised when two providers for the same name have been instantiated.

`syringe.get(name)`

Returns the provided instance.

Parameters `name` (*str*) – The lookup name.

Returns The provided instance.

class `syringe.inject(name)`

Bases: `object`

Injects the instance of a provided class to the current context.

`syringe.mock(name, m=None)`

Provide a mock overriding any namespace conflicts.

If a second argument is passed, it is provided using the specified name. Otherwise a new `unittest.mock.Mock` instance is provided.

Parameters

- **name** (*str*) – The name of the mock to provide.
- **m** – The optional object to provide as a mock.

`syringe.provides(name)`

Provides the decorated class as an injectable object.

The class `__init__` method is monkey patched so a `DuplicateProviderError` is raised when another instance is created for this name.

Parameters `name` (*str*) – The name under which the provided class will be accessible.

exception `syringe.NoCandidateError`

Bases: `exceptions.KeyError`

Raised when no candidate has been registered for the requested name.

S

`syringe`, 3

C

`clear()` (in module `syringe`), 4

D

`DuplicateProviderError`, 4

G

`get()` (in module `syringe`), 4

I

`inject` (class in `syringe`), 4

M

`mock()` (in module `syringe`), 4

N

`NoCandidateError`, 4

P

`provides()` (in module `syringe`), 4

S

`syringe` (module), 1