

## 概述

本文主要分享如何将pytest和requests结合在一起使用，让大家有个初步的了解。

主要内容有：

- pytest简介
- pytest + requests示例

## 测试接口

你可以从以下链接获取到我们实例中用到的接口：

```
http://jsonplaceholder.typicode.com
```

注意: 因该服务在海外，请勿持续请求该服务的接口。

如果你有需要可以下载对应的服务包，部署到本地来进行测试。

## pytest介绍

pytest是python的一个测试库，它使得测试更为简单，它拥有许多高级的功能和插件，有助于更高级的测试解决方案。

安装pytest，命令如下：

```
pip install pytest
```

下面我们来看一个pytest的基本示例，体验下它的魅力：

```
#-*- coding:utf-8 -*-

__author__ = "苦叶子"

import pytest

# 测试加减法
class TestDemo:
    # 加法
    @pytest.mark.parametrize("a, b, expected",
                             [(1,2,3), (2,3,5), (3,4,8)])
    def test_add(self, a, b, expected):
        # 求和
        sum = a + b

        # 断言
        assert sum == expected

# 减法
```

```

@pytest.mark.parametrize("a, b, expected",
                           [(1,2,-1), (8,3,5), (3,4,8)])
def test_sub(self, a, b, expected):
    # 减法
    s = a - b

    # 断言
    assert s == expected

```

将上述代码保存至test\_requests\_pytest\_demo.py中，在该文件所在的目录执行下述命令：

```
pytest -q
```

执行结果如下图：

```

D:\writing\Python3接口测试>pytest -q
..F..F
===== FAILURES =====
_____ TestDemo.test_add[3-4-8] _____
self = <test_requests_pytest_demo.TestDemo object at 0x00000000040F9AC8>, a = 3
b = 4, expected = 8

@pytest.mark.parametrize("a, b, expected",
                           [(1,2,3), (2,3,5), (3,4,8)])
def test_add(self, a, b, expected):
    # 求和
    sum = a + b

    # 断言
    assert sum == expected
E       assert 7 == 8
test_requests_pytest_demo.py:17: AssertionError
_____ TestDemo.test_sub[3-4-8] _____
self = <test_requests_pytest_demo.TestDemo object at 0x00000000041A8630>, a = 3
b = 4, expected = 8

@pytest.mark.parametrize("a, b, expected",
                           [(1,2,-1), (8,3,5), (3,4,8)])
def test_sub(self, a, b, expected):
    # 减法
    s = a - b

    # 断言
    assert s == expected
E       assert -1 == 8
test_requests_pytest_demo.py:27: AssertionError
2 failed, 4 passed in 0.20 seconds

```

说明：

@pytest.mark.parametrize 为pytest参数化用法

## 示例

下面我们简单的演示下pytest和requests结合应用的示例

```

#-*- coding:utf-8 -*-

__author__ = "苦叶子"

import pytest
import requests

# 基于pytest requests测试 接口

```

```

class TestRequestDemo:
    # 初始化
    url = "http://jsonplaceholder.typicode.com"
    session = requests.session()

    # 测试获取所有用户信息接口
    def test_get_posts(self):
        r = self.session.get(self.url + "/posts")

        # 断言状态码
        assert r.status_code == 200

        # 断言响应头信息
        assert r.headers["Content-Type"] == "application/json; charset=utf-8"

        # 断言用户总数
        assert len(r.json()) == 100

    # 测试获取指定用户信息接口
    def test_get_posts_by_id(self):
        r = self.session.get(self.url + "/posts/1")

        # 断言状态码
        assert r.status_code == 200

        # 断言响应头信息
        assert r.headers["Content-Type"] == "application/json; charset=utf-8"

        # 验证用户id
        data = r.json()
        assert data["userId"] == 1

    # 测试删除指定用户信息接口
    def test_delete_posts_by_id(self):
        r = self.session.delete(self.url + "/posts/1")

        # 断言状态码
        assert r.status_code == 200

        # 断言响应头信息
        assert r.headers["Content-Type"] == "application/json; charset=utf-8"

```

将上述代码保存至test\_requests\_demo.py中，在该文件所在目录执行下述命令：

```
pytest -q
```

执行结果入下图：

```
D:\writing\Python3接口测试\demo>pytest -q
...
3 passed in 2.54 seconds
```

扫一扫关注微信公众号：

