

request+pytest框架实现接口自动

一、pytest框架是什么？

成熟的全功能的python测试框架（框架，字面意思，类似高楼大厦的框架结构）

- 1、简单灵活，容易上手，支持随机/文件参数化、断言
- 2、支持单元测试、接口测试、web方面的UI自动化测试
- 3、可以兼容其他第三方库，比如pytest-html、allure

二、pytest框架的思路

- 1、将公关方法、配置文件，封装成函数，或者是类
- 2、将测试用例封装成函数/类
- 3、需要使用到读取测试数据

三、pytest框架前提与划分

- 1、安装pytest，安装pytest-html

- 2、pytest框架的模块划分：6个模块

common：存放公关方法、元素定位方法、数据操作、驱动方法

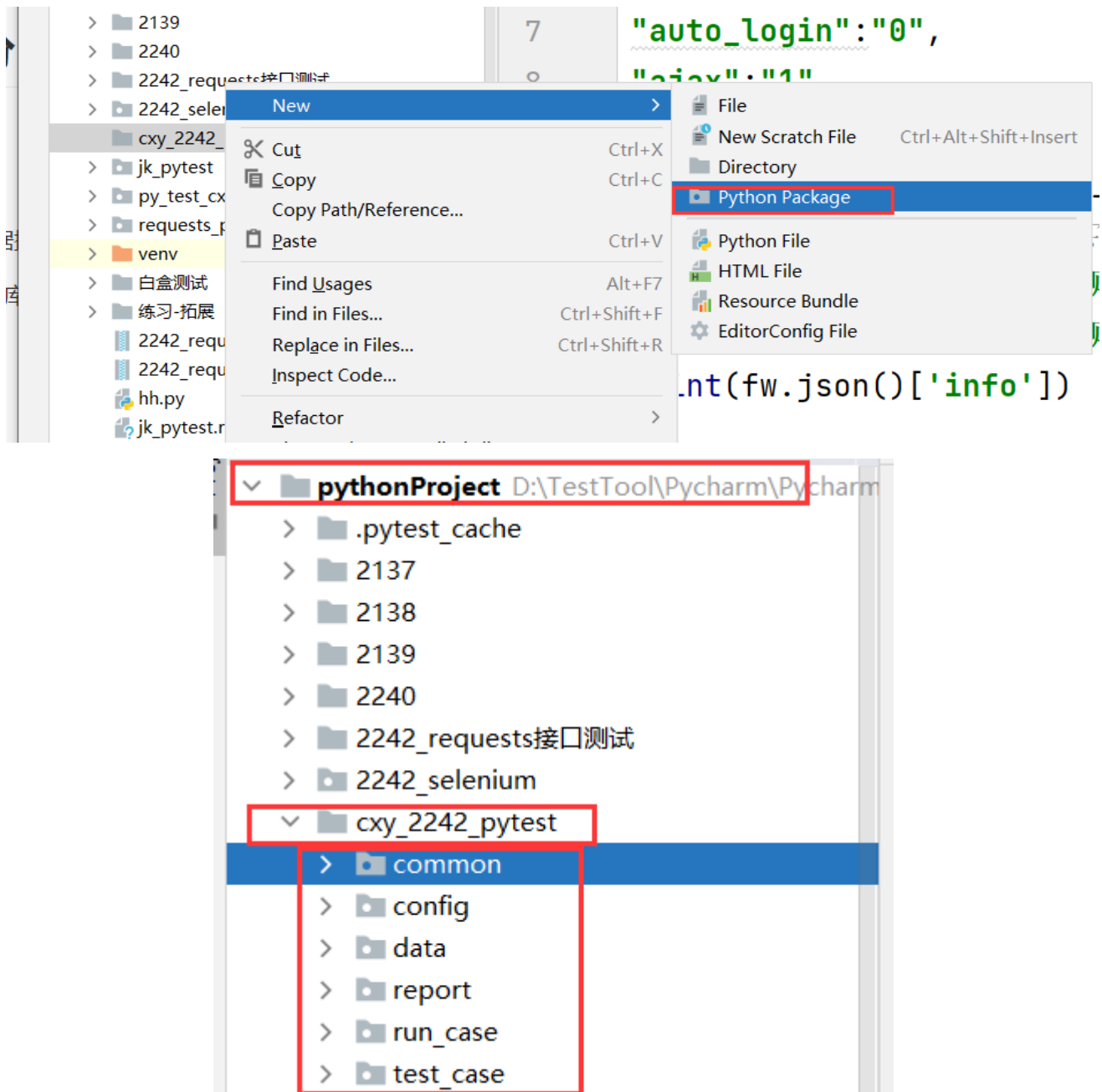
config：配置文件，存放配置、IP，域名，数据库的配置

data：数据层，用于测试数据

report：生成测试报告存放的目录

run_case：执行测试用例主程序的目录

test_case：存放测试用例



注意：1、test_case目录下的python文件，需要以test开头（结尾）

2、test_case目录下的python文件中的函数/类方法，也是需要以test开头（结尾）

四、pytest框架的封装操作步骤

1、test_case目录下创建 fw_login.py 文件 （先不加test开头）

把之前方维登陆断言成功的用例复制过来

The screenshot shows the PyCharm IDE with a project named '2242_selenium'. The file explorer on the left highlights the 'fw_login.py' file. The main editor displays the following Python code:

```
1 import requests
2 fw_url = "http://106.52.182.140/fanwe/index.php?ctl=user&act=doLogin&"
3 fw_data = {
4     "email": "test02",
5     "user_pwd": "UEJHb3BWWmhuVEZ0VEVtbE1McnlRUVZGeExnYXFPdk1FaW1hWXBMcE9kS"
6     "auto_login": "0",
7     "ajax": "1"
8 }
9 fw = requests.post(url=fw_url, data=fw_data)
10 assert '本站需绑定第三方托管账户' in fw.json()['info'], "登陆失败"
11
```

The Run window at the bottom shows the command: `D:\TestTool\Python\python.exe D:/TestTool/Pycharm/PycharmProjects/pythonProject/cxy_2242_pytest` and the output: `Process finished with exit code 0`.

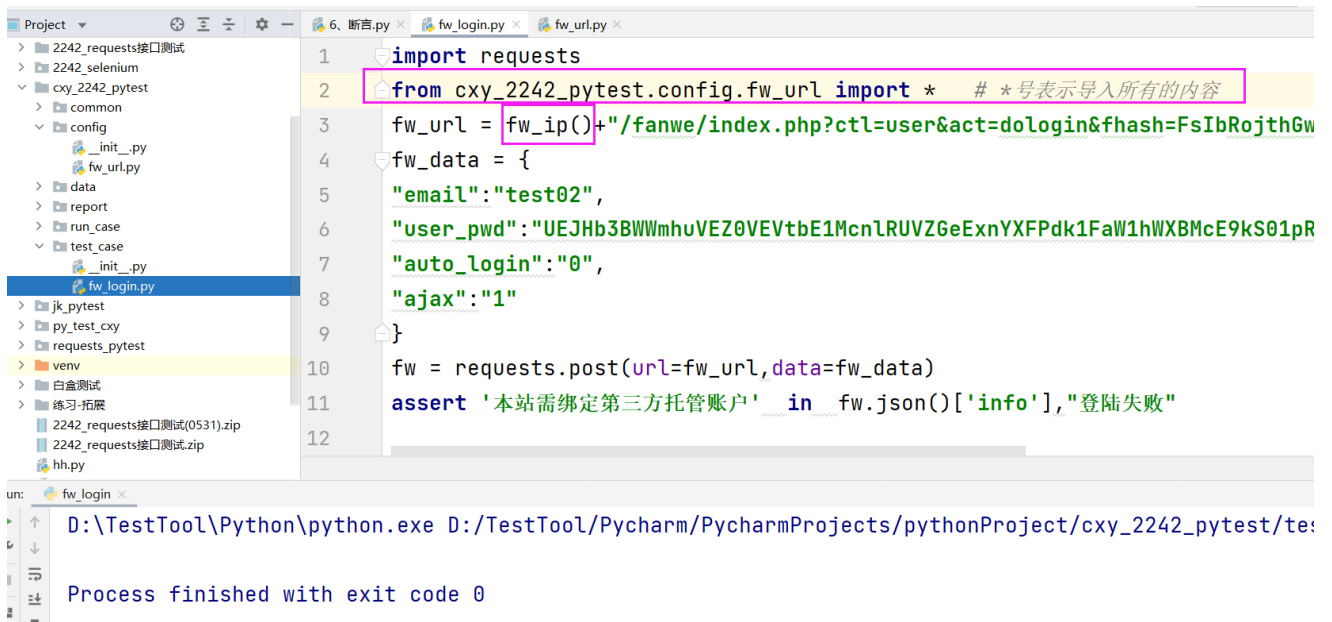
2、为了方便后续维护，切换脚本运行环境，把IP地址拿出来单独封装，统一调用

后续更换环境时，直接切换IP的注释即可

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left highlights the 'fw_url.py' file. The main editor displays the following Python code:

```
1
2 def fw_ip():
3     ip = "http://106.52.182.140" # 测试环境
4     # ip = "http://101.33.250.42" # 研发环境=开发环境
5     return ip
6
7
8 # print(fw_ip()) # 打印到控制台进行调试查看，完成后，需要删除或注释掉
9
```

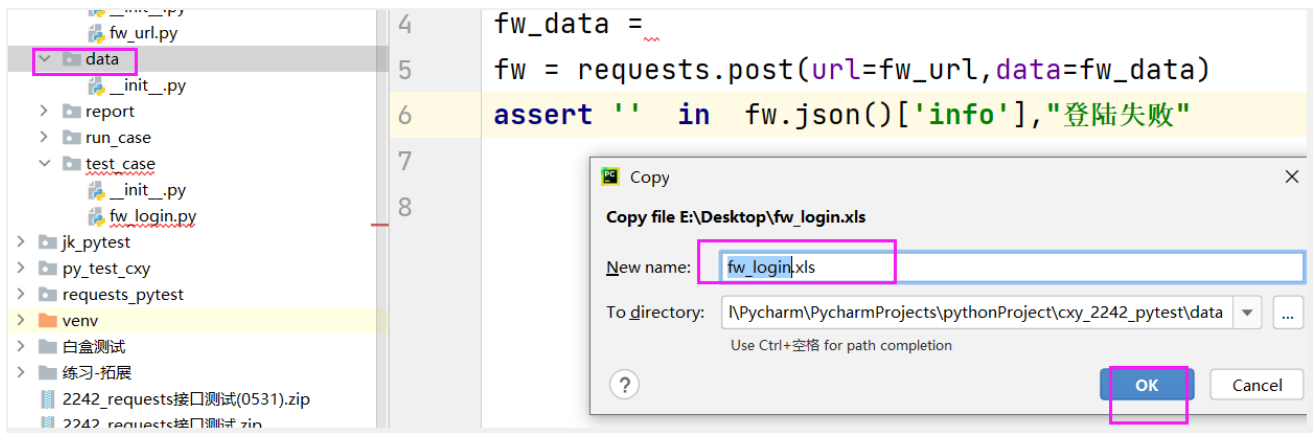
3、封装好后，进行调用，调用需要先导包，使用from。。。导包



此时，执行测试用例，依旧是可以执行通过，断言成功。

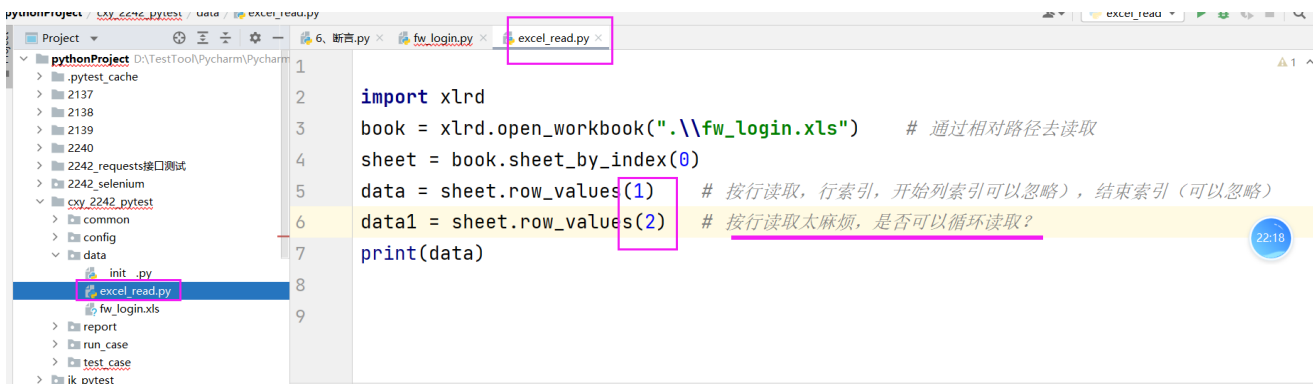
4、将测试用例使用.xls表格文件储存到data目录下

	A	B	C	D
1	请求路径	请求正文	预期结果	备注
	/fanwe/index.php?ctl=user&act=dologin&fhash=FslbRojthGwgtekQVcJmfqgznrZQNfsSjzcHEInjByLLVOwOI	{ "email": "test02", "user_pwd": "UEJHb3BWWmhuVEZ0VEVtbE1McnlRUVZGeExnYXFPdk1FaW1hWXBMcE9kS01pRVFyclEIMjV1NjVCOSUyNXU3RUY0cXdlMTIzNDU2JTl1dThGNkYIMjV1NEVGNg==", "auto_login": "0", "ajax": "1" }	本站需绑定第三方托管账户	正向
2	/fanwe/index.php?ctl=user&act=dologin&fhash=FslbRojthGwgtekQVcJmfqgznrZQNfsSjzcHEInjByLLVOwOI	{ "email": "test23567", "user_pwd": "UEJHb3BWWmhuVEZ0VEVtbE1McnlRUVZGeExnYXFPdk1FaW1hWXBMcE9kS01pRVFyclEIMjV1NjVCOSUyNXU3RUY0cXdlMTIzNDU2JTl1dThGNkYIMjV1NEVGNg==", "auto_login": "0", "ajax": "1" }	用户不存在	反向
3	/fanwe/index.php?ctl=user&act=dologin&fhash=FslbRojthGwgtekQVcJmfqgznrZQNfsSjzcHEInjByLLVOwOI	{ "email": "test02", "user_pwd": "UEJHb3BWCnlRUVZGeExnYXFPdNEVGNg==", "auto_login": "0", "ajax": "1" }	密码错误	反向
4	没有引号			
5		大括号、双引号，最后键值对的逗号		
6				
7				

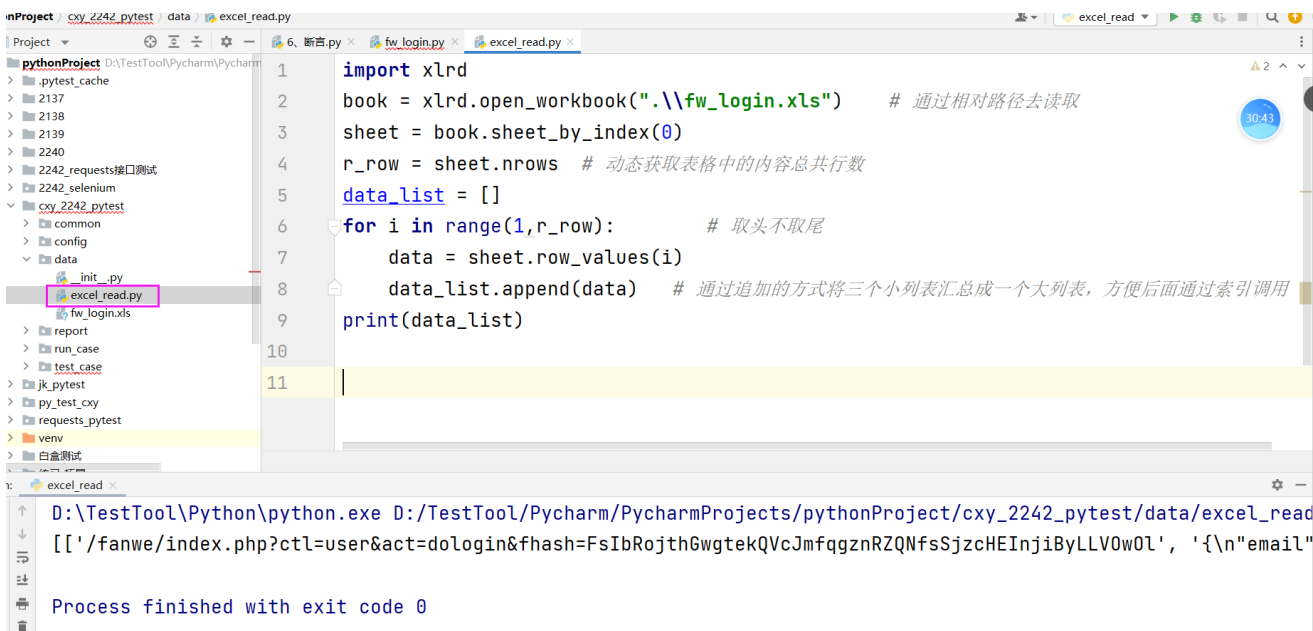


5、通过xlrd进行读取表格数据

按行读取太麻烦

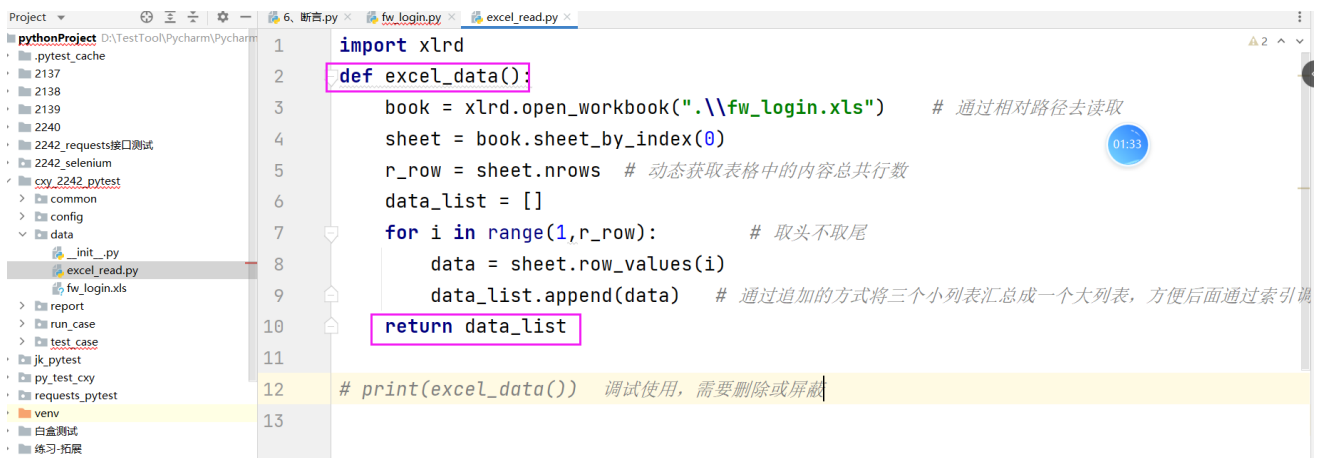


更换成循环读取, 并使用append追加到一个大列表中, 方便后面通过索引调用



6、读取测试数据封装成函数, 方便调用

注意函数需要使用return



```
1 import xlrd
2 def excel_data():
3     book = xlrd.open_workbook(".\\fw_login.xls") # 通过相对路径去读取
4     sheet = book.sheet_by_index(0)
5     r_row = sheet.nrows # 动态获取表格中的内容总共有行数
6     data_list = []
7     for i in range(1, r_row): # 取头不取尾
8         data = sheet.row_values(i)
9         data_list.append(data) # 通过追加的方式将三个小列表汇总成一个大列表，方便后面通过索引调
10    return data_list
11
12 # print(excel_data()) 调试使用，需要删除或屏蔽
13
```

7、调用测试数据

使用装饰器将数据 循环 依次 读取回来，并使用 公式：@pytest.mark.parametrize("字符data",函数变量) 表示把 函数变量的数据依次循环 赋值给 字符data 大列表中的三个小列表，依次给到data

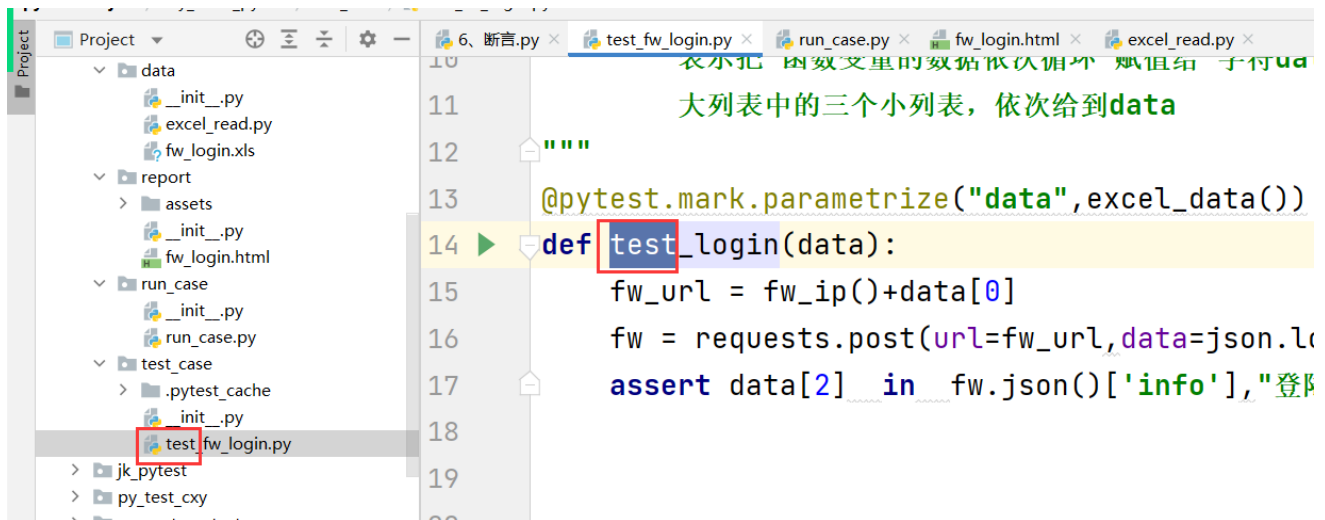
注意：测试用例名需要使用test_开头



```
1 import requests
2 from cxy_2242_pytest.config.fw_url import * # *号表示导入所有的内容
3 from cxy_2242_pytest.data.excel_read import *
4 import pytest
5
6 """
7 装饰器
8 使用装饰器将数据 循环 依次 读取回来，并使用
9 公式：@pytest.mark.parametrize("字符data",函数变量)
10 表示把 函数变量的数据依次循环 赋值给 字符data
11 大列表中的三个小列表，依次给到data
12 """
13 @pytest.mark.parametrize("data", excel_data())
14 def test_login(data):
15     fw_url = fw_ip()+data[0]
16     fw = requests.post(url=fw_url, data=data[1])
17     assert data[2] in fw.json()['info'], "登陆失败"
```

8、执行pytest框架测试用例，各种报错及解决方法

报 no test ，是没找到测试用例，不是路径问题，就是 测试脚本没有使用test开头，测试用例函数方法没有使用test开头：



报找不到excel表格数据

```

file_format = inspect_format(filename, file_contents)
C:\Users\admin\AppData\Roaming\Python\Python310\site-packages\xlrd\__init__.py:60:
    with open(path, "rb") as f:
E   FileNotFoundError: [Errno 2] No such file or directory: '..\\fw_login.xls'
collected 0 items / 1 error

!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!
===== 1 error in 0.23s =====

```

在当前目录找不到

因为是从测试用例test_fw_login.py去执行，需要将读取表格的路径增加多一层路径



```

file_format = inspect_format(filename, file_contents)
C:\Users\admin\AppData\Roaming\Python\Python310\site-packages\xlrd\__init__.py:60: in inspect_f
    with open(path, "rb") as f:
FileNotFoundError: [Errno 2] No such file or directory: '..\\fw_login.xls'
collected 0 items / 1 error

!!!!!!!!!!!!!!!!!!!! Interrupted: 1 error during collection !!!!!!!!!!!!!!!!!!!!!!!

```

报JSON格式问题是因为通过文件参数化读取回来的数据是str类型，需要进行转换json格式

```
12 """
13 @pytest.mark.parametrize("data", excel_data())
14 def test_login(data):
15     fw_url = fw_ip()+data[0]
16     fw = requests.post(url=fw_url, data=json.loads(data[1]))
17     assert data[2] in fw.json()['info'], "登陆失败"
18     test_login()
```

Tests failed: 3 of 3 tests - 176 ms

```
> raise JSONDecodeError("Expecting value", s, err.value) from None
E json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
D:\TestTool\Python\lib\json\decoder.py:355: JSONDecodeError
```

非JSON格式，需要转换

9、生成报告与查看报告

```
pythonProject > cxy_2242_pytest > run_case > run_case.py
1 import pytest
2 if __name__ == '__main__':
3     pytest.main(["..\\test_case\\test_fw_login.py", "--html=..\\report\\fw_login.html"])
4 """
5 方法一、pytest.main(["测试用例的路径", "--html=路径\\报告名称.html"])
6 方法二、pytest.main(["测试用例的路径\\用例文件", "--html=路径\\报告名称.html"])
7 """
8
9
```

```
pythonProject > cxy_2242_pytest > report > fw_login.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8"/>
5 <title>Test Report</title>
6 <link href="assets/style.css" rel="stylesheet" type="text/css"/></head>
7 <body onload="init()">
8 <script>/* This Source Code Form is subject to the terms of the Mozilla Public
9 * License, v. 2.0. If a copy of the MPL was not distributed with this file,
10 * You can obtain one at http://mozilla.org/MPL/2.0/. */
11
12
13 function toArray(iter) {
14     if (iter === null) {
```