

基于 python3 的接口自动化测试框架说明

浪晋

一、为什么开发这个框架

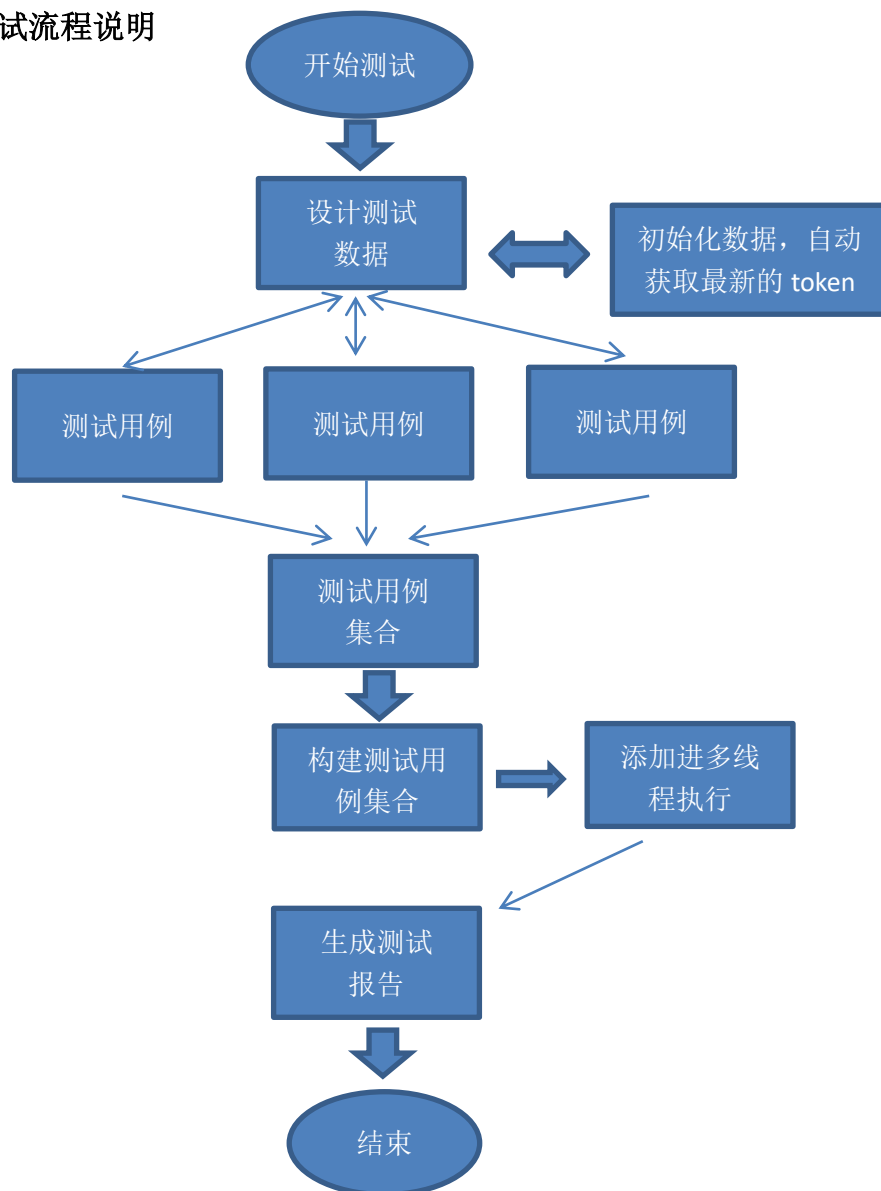
随着公司接口的增多，开发一边开发，测试一边测试，导致了接口的不稳定。为了去验证接口是否正常，耗费了越来越多的人力，而且大量的都是重复的工作。

为了减轻测试的压力，所以开发了这个轻量级的接口自动化测试框架。

二、自动化测试框架介绍。

- a) 基于 python 脚本编写，不需要借助其他工具，环境配置简单。
- b) 采用数据驱动测试的方式，后期仅仅只需要维护一个测试数据的 EXCLE 表格。
- c) 测试用例可以分模块编写，方便对用例的管理维护。
- d) 测试用例和测试执行分离，实现执行测试时用例可以任意配置。
- e) 实现多线程执行，可以同时运行多个测试集。
- f) 测试结束后直接生成 EXCLE 报告，统计测试执行情况，执行情况详细，能够快速定位问题，并且容易扩展优化。

三、自动化测试流程说明



1、测试数据

在 testdata.xls 表格中编写每一条测试用例的测试数据。

注意：目前代码版本不支持.xlsx 格式的表格。

| | | | | | | |
|-------------------------|----------|-----------|----------|--------|------|-----|
| 添加管理员 | | | | | | |
| test_1add_admin() | | | | | | |
| userName | realName | status | password | remark | code | msg |
| jintest | jintest | 1 | 123456 | 测试 | 400 | |
| jintest | jintest | 1 | 123456 | 测试 | 400 | |
| jintest | jintest | 1 | 123456 | 测试 | 400 | |
| jintest | jintest | 1 | 123456 | 测试 | 200 | |
| jintest1111111111111111 | jintest | 1 | 123456 | 测试 | 400 | |
| | | | | | | |
| 删除管理员 | | | | | | |
| test_2del_admin() | | | | | | |
| id | code | msg | | | | |
| 209 | 200 | | | | | |
| 210 | 200 | | | | | |
| 211 | 200 | | | | | |
| | | | | | | |
| 修改管理员 | | | | | | |
| test_3edit_admin() | | | | | | |
| id | realName | adminType | remark | status | code | msg |
| 40 | aaa | 0 | 测试 | 1 | 200 | |
| 41 | XXX | 0 | 测试 | 1 | 200 | |
| 42 | eee | 0 | 测试 | 1 | 200 | |
| 43 | hhh | 0 | 测试 | 1 | 200 | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

2、自动获取 token

获取初始 token，并自动写入 testdata 中。

```
r = requests.post(hurl+'/Login', data=json.dumps(hdata), headers=headers)
hjson = json.loads(r.text) #获取并处理返回的json数据
hcode = str(hjson['code'])
print('请求返回状态为: '+hcode)
if hcode == table.cell(9,1).value:
    token = hjson['data']['token'] #获取token
    print('当前token为: '+token)
    #将获取的TOKEN保存到testdata中
    oldWb = xlrd.open_workbook('C:/Jin/workpase/ApiTest/src/testdata.xls',f
    newWb = copy(oldWb)
    newWs = newWb.get_sheet(0)
    newWs.write(8, 1, token)
    print("Token写入成功")
    newWb.save('C:/Jin/workpase/ApiTest/src/testdata.xls')
    print("TestdDate保存成功")
else:
    print('登陆失败，程序退出')
    exit()
```

3、自动写入 token 到测试数据中

获取到的 token 写到 testdata 中，方便其他测试用例调用，保证每次都是调用的最新的 token。

| A | B |
|--------------|----------------------------------|
| 初始化数据 | |
| 账号 | test22 |
| 密码 | 123456 |
| otp | 123456 |
| content-type | application/json;charset=UTF-8 |
| url | http://10.254.254.46:31501 |
| token | 8403a1b1477337c43fa47df060f5201a |
| code | 200 |

自动获取并写入的 token，保证其他用例每次调取的都是最新的 token

4、编写测试用例

测试用例脚本中，所有数据都从 testdata.xls 中读取。

```
def test_1add_admin():
    global hhhdata
    for i in range(3,8):
        table = Testdata.sheets()[1]#选择excel表中的sheet
        username = table.cell(i,0).value#读取测试数据
        realName = table.cell(i,1).value
        status = table.cell(i,2).value
        password = table.cell(i,3).value
        remark = table.cell(i,4).value
        data = {
            "username": username,
            "realName": realName,
            "password": password,
            "remark": remark,
            "status": status}

        headers = {
            'content-type': hcontent_type,
            'token': htoken
        }
```

5、构建测试集合

导入测试用例，把所有需要执行的用例放到一起。

```
import GetToken
import AdminTestCase_01#导入测试用例

def HRunner():
    global TestReportData
    print('开始测试')
    GetToken.test_get_token()
    AdminTestCase_01.test_1add_admin()#添加管理员
    AdminTestCase_01.test_2del_admin()#删除管理员
    # print(AdminTestCase_01.hlist)
    TestReportData = AdminTestCase_01.hlist#返回测试结果
    print(TestReportData)
    print('结束测试')
    return TestReportData
```

6、执行测试用例集

执行配置好了的测试用例集合并生成测试报告。

一直下一步，直到出现这个界面。添加 `add python.exe to path`



然后继续下一步，安装完成。

最后打开 cmd，输入 python，若正确执行则安装成功。

如图。

```
C:\Users\admin>python
Python 3.5.3 (v3.5.3:1880cb95a742, Jan 16 2017, 16:0
Type "help", "copyright", "credits" or "license" for
>>> _
```

2、安装脚本中需要的库

安装方法，python 安装成功后自带了 pip 的工具，打开 cmd

输入 pip install 库名

如图，安装 lxml 库

```
C:\Users\admin>pip install lxml
```

安装成功后，可输入 pip list 查看已安装的库。

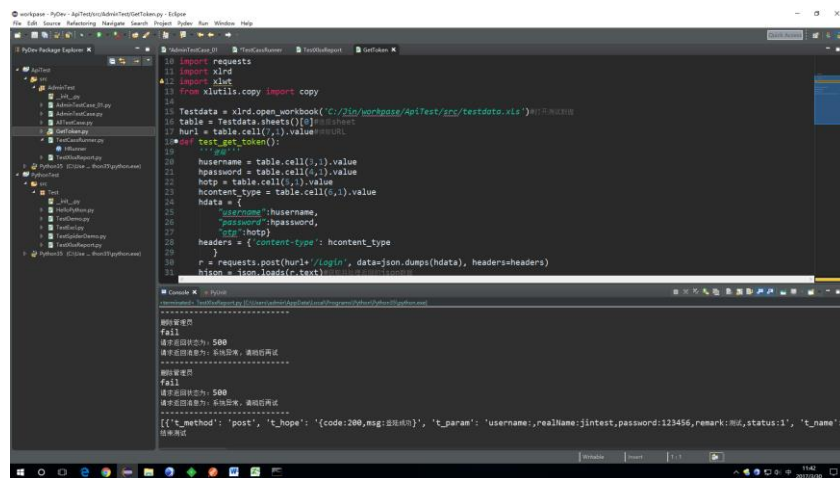
下图为接口自动化测试框架所依赖的库。

```
C:\Users\admin>pip list
DEPRECATION: The default format will switch to columns in the future. You
can avoid this warning by passing --format=(legacy|columns) in your pip.conf under the [list] section to
beautifulsoup4 (4.5.3)
certifi (2017.7.27)
chardet (3.0.4)
lxml (3.7.3)
pip (9.0.1)
requests (2.13.0)
setuptools (28.8.0)
xlrd (1.0.0)
xlwt (1.2.0)
C:\Users\admin>
```

3、选择自己喜欢的 IDE

环境配置就搞定了，大家只需要选择自己喜欢的 IDE 进行编码即可。

如图：



五、持续优化

- 1、接口请求失败无响应超时的情况。
- 2、程序异常中断的情况。
- 3、封装测试用例脚本内的重复方法。

六、作者的废话

测试报告是网上找的方法自己改了改，不是自己写的。

给这个还不成熟的框架取了个名字叫 FreeTestGo.

各个脚本的作用看脚本里 header 里的注释。

testdata 那个表格就是写测试数据的地方，脚本里的路径记得修改成你自己的。

目前正在做的工作，把数据库的方法写进用例和预期结果做对比。

持续优化脚本，尽量的做到只用维护测试数据，不对脚本进行操作。

未来的目标，集成 jenkins，把测试数据都写进数据库，做成可以直接通过 web 写到数据库里，做成一个大的成熟的东西。可是任重道远啊。估计也完不成这个目标。

呵呵呵

有问题可以联系我。

QQ: 719822330.

微信也是这个。

邮箱: fenyukuang@163.com

微博: __浪晋

作者: 浪晋