# 使用requests做接口测试

编写: 邱麒燃

时间: 2018年11月26号

这里主要介绍接口自动化测试,其中会涉及到Requests 库、unittest 单元测试框架,MySQL 数据库的操作,HTMLTestRunner 生成测试报告,以及发送邮件。

## 环境准备

• 安装python

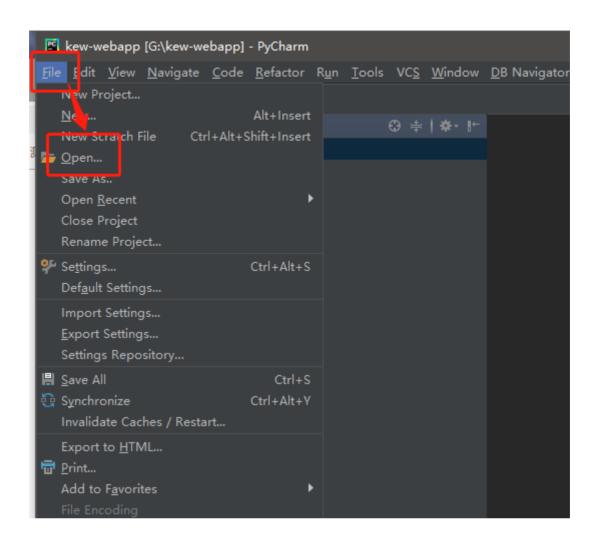
下载地址: https://www.python.org/downloads/



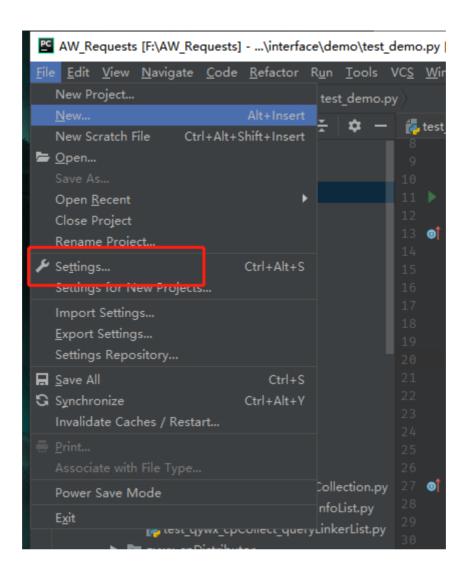
• 安装pycharm

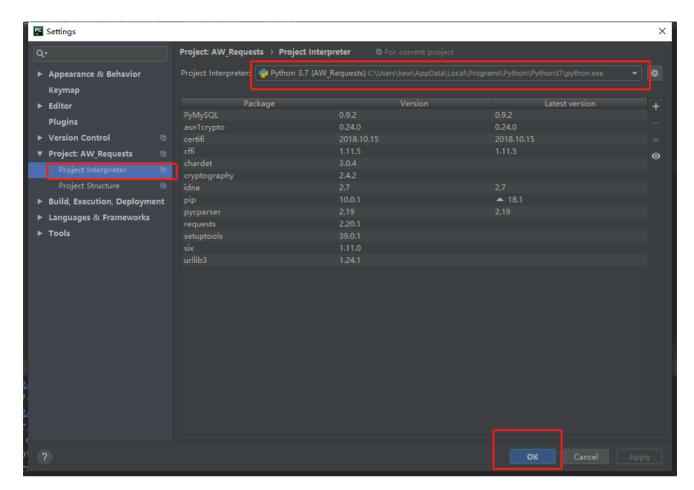
下载地址: https://www.jetbrains.com/pycharm/

导入AW接口框架



配置,把安装的python文件地址导入





### • 安装requests

pip install requests

中文文档: http://cn.python-requests.org/zh CN/latest/

• 安装数据库连接

pip install PyMySQL

在pycharm下方的terminal安装request和PyMySQL

```
<u>F</u>ile <u>E</u>dit <u>V</u>iew <u>N</u>avigate <u>C</u>ode <u>R</u>efactor R<u>u</u>n <u>T</u>ools VC<u>S W</u>indow <u>H</u>elp
AW Requests \ interface \ interface \ demo \ is test demo.py
                                 ■ Project ▼
1: Project
  ▼ Marken AW Requests F:\AW Requests
    ▼ 🖿 framework
         💤 base.py
                                                        class Demo(unittest.TestCase):
         # HTMLTestRunner.py
                                                             def setUp(self):
    ▼ Iinterface
       ▶ □ арі
       ▼ 🖿 demo
            ke.xls
            test_demo.py
                                                                 #获取公共接口类
       ▶ miniapp
                                                                 self.base = Base()
                                                                 #当前测试接口地址
url = '/dashboard/data'
       ▼ 🖿 qywx
         ▶ 🖿 gywx account
         ▶ 🖿 qywx_brokerVip
         ▶ 🖿 qywx_common
         ▼ 🖿 qywx_cpCollect
                                                                 self.db = Base().mysql_connect()
              🛵 _init_.py
              test_qywx_cpCollect_linkerCollection.py
                                                             def tearDown(self):
              test_qywx_cpCollect_queryInfoList.py
                                                                 self.db.close()
              test_qywx_cpCollect_queryLinkerList.py
         ▶ □ qywx_cpDistributor
                                                             def test_01(self):
         ▶ □ qywx cpIM
                                                                 ///用例1//
#获取token
         ▶ □ qywx_cpInformation
                                                                 Authorization = self.base.Authorization()
         ▶ 🖿 qywx_cpInformationCollect
                                                                 headers = {'Authorization':Authorization}
         ▶ 🖿 qywx_cpInformationShare
        Using cached https://files.pythonhosted.org/packages/ea/cd/35485615f45f30a510576f1a56d1e0a7ad7bd8al
   Collecting pycparser (from cffi!=1.11.3,>=1.7->cryptography->PyMySQL)
        Using cached https://files.pythonhosted.org/packages/68/9e/49196946aee219aead1290e00d1e7fdeab856778
🚹 🔃 Structure
      Installing collected packages: six, pycparser, cffi, asn1crypto, cryptography, PyMySQL
        Running setup.py install for pycparser ... done
      Successfully installed PyMySQL-0.9.2 asn1crypto-0.24.0 cffi-1.11.5 cryptography-2.4.2 pycparser-2.19
      You are using pip version 10.0.1, however version 18.1 is available.
¥ 2: Favorites
      You should consider upgrading via the 'python -m pip install --upgrade pip' command.
      F:\AW Requests>

☐ Terminal

                                              Python Console
```

## 自动化框架实现

#### AW\_Requests 框架

1. framework/: 公共类文件夹。
2. interface/: 接口文件夹。
3. report/: 报告文件夹。
4. config.ini: 配置文件。
5. run\_tests.py: 执行入口。

## 写接口

get例子

```
#参数
payload = {'key1': 'value1', 'key2': 'value2'}
#请求头
headers = {'user-agent': 'my-app/0.0.1'}
#请求
r = requests.get("http://xxxxx", params=payload,headers=headers)
```

post例子

```
#参数

payload = {'key1': 'value1', 'key2': 'value2'}

#请求头

headers = {'user-agent': 'my-app/0.0.1'}

#请求

r = requests.post("http://xxxxx", params=payload,headers=headers)
```

## json格式

```
1 + {
2 +
             "data": [
                      "classify": "user_head_img", 
"createId": "1",
  4
                                                                            ['data'][0]['classify']
  5
                      "createTime": "2018-08-16 22:50:10",
  6
                      "id": "136",
"itemCode": "http://720ljq2test-10037467.file.myqcloud.com/ljqzs/user_head_img/male_001.png",
"itemName": "头像男001",
  8
                      "status": 1,
"updateTime": "2018-08-16 22:50:10"
 11
 12
 13
 14 •
                      "classify": "user_head_img",
"createId": "1",
 15
                                                                                 ['data'][1]['classify']
 16
                      "createTime": "2018-08-16 22:50:10",
 17
                      "id": "137",
"itemCode": "http://7201jq2test-10037467.file.myqcloud.com/ljqzs/user_head_img/male_002.png",
"itemName": "头像男002",
 19
 20
 22
23
24
                      "status": 1,
"updateTime": "2018-08-16 22:50:10"
 25 ▶
 36 ►
47 ►
                  ₹<mark>6</mark>3},
 58 ▶
 69 ⊧
                  {□},
 80 +
 91 ▶
113 ▶
124 ⊦
135 •
146 +
                  ⟨∰⟩,
157 ▶
168 +
                 {□}},
179 ▶
190
             "handlerTime": "1543284498659",
191
            "msg": "正常",
"result": true,
192
193
            "returnCode": 10000,
194
195
            "timezone":
196 }
```

## 断言

```
self.assertEqual(a,b) #判断a和b是否一致
self.assertNotEqual(a,b) #判断a和b是否不一致
```

## 例子

1. 获取默认头像列表:/qywx/dictionary/getAgentHeadImgList

```
def test_01(self):
    '''查询经纪人默认头像列表'''
    #获取token
    Authorization = self.base.Authorization()
    headers = {'Authorization': Authorization}
    #请求
    r = requests.get(self.test_url,headers=headers)
    #断言状态码
```

```
self.assertEqual(r.status_code, 200)
#打印json数据
print(r.json())
#断言msg
self.assertEqual(r.json()['msg'], '正常')
#断言result
self.assertEqual(r.json()['result'], True)
#断言returnCode
self.assertEqual(r.json()['returnCode'], 10000)
#断言返回的数据
self.assertEqual(r.json()['data'][0]['classify'],'user_head_img')
self.assertEqual(r.json()['data'][0]['itemName'],'头像男001')
print('/qywx/dictionary/getAgentHeadImgList 查询经纪人默认头像列表 ok')
```

2. 编辑经纪人信息:/qywx/user/updateAgentByUserId

```
def test_01(self):
    '''修改经纪人信息 '''
   #获取token
   Authorization = self.base.Authorization()
   headers = {'Authorization': Authorization}
   #定义参数
   agentId = self.base.agentId()
   name = '小比利'
   wechatAccount= '15622518977'
   majorRegion= '广东省/深圳市/南山区'
   distributorId= '120'
   institutionId= '42'
   storeId= '216'
   signature= '这里市个性签名 (名片介绍) '
   tempPhone= '15622518977'
   lableList = "[{'id': 0,'labelId': 0,'labelName': 'string','userId': 0}]"
   params = {'agentId': agentId,'name':name,'wechatAccount':wechatAccount,
              'majorRegion':majorRegion,
              'distributorId':distributorId,'institutionId':institutionId,
              'storeId':storeId,
              'signature':signature,'tempPhone':tempPhone,'lableList':lableList}
   #请求
   r = requests.put(self.test_url, data=params, headers=headers)
   self.assertEqual(r.status_code, 200)
   print(r.json())
   self.assertEqual(r.json()['result'], True)
   self.assertEqual(r.json()['returnCode'], 10000)
   self.assertEqual(r.json()['msg'], '正常')
   self.assertEqual(r.json()['data']['agentId'], agentId)
   self.assertEqual(r.json()['data']['name'], name)
   self.assertEqual(r.json()['data']['wechatAccount'], wechatAccount)
```

```
self.assertEqual(r.json()['data']['majorRegion'], majorRegion)
self.assertEqual(r.json()['data']['distributorId'], distributorId)
self.assertEqual(r.json()['data']['institutionId'], institutionId)
self.assertEqual(r.json()['data']['storeId'], storeId)
self.assertEqual(r.json()['data']['signature'], signature)
self.assertEqual(r.json()['data']['tempPhone'], tempPhone)
self.assertEqual(r.json()['data']['lableList'], lableList)
print('/qywx/user/updateAgentByUserId 修改经纪人信息 ok')
```

#### 3.查询数据库信息

```
def setUp(self):
    # 连接数据库
    self.db = Base().mysql_connect()
```

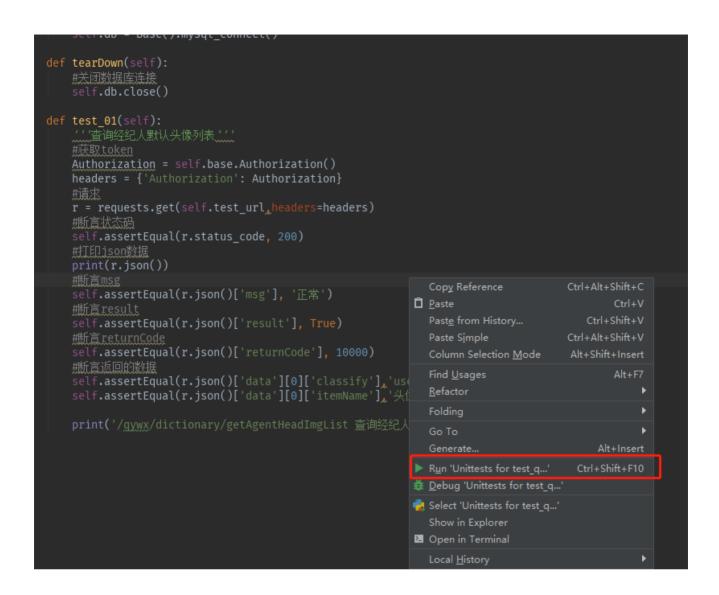
```
def test_05(self):
    #语句
    sql = 'SELECT linker_id FROM helper_720_project WHERE subscribe_user_id =

4477;'

#创建游标
    cursor = self.db.cursor()
    #执行
    cursor.execute(sql)
    #获取数据
    data = cursor.fetchall()
    #赋值数据变量
    linkerId = data[0][0]
    #断言json返回的数据与数据库是否一致
    self.assertEqual(r.json()['data']['records'][0]['linkerId'], linkerId)
    #关闭数据库
    self.db.close()
```

## 运行查看结果

1. 单个文件运行,右键run即可



2.运行所有,打开根目录下run\_test.py,运行即可,报告在report查看。