

使用requests做接口测试

编写：邱麒麟

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

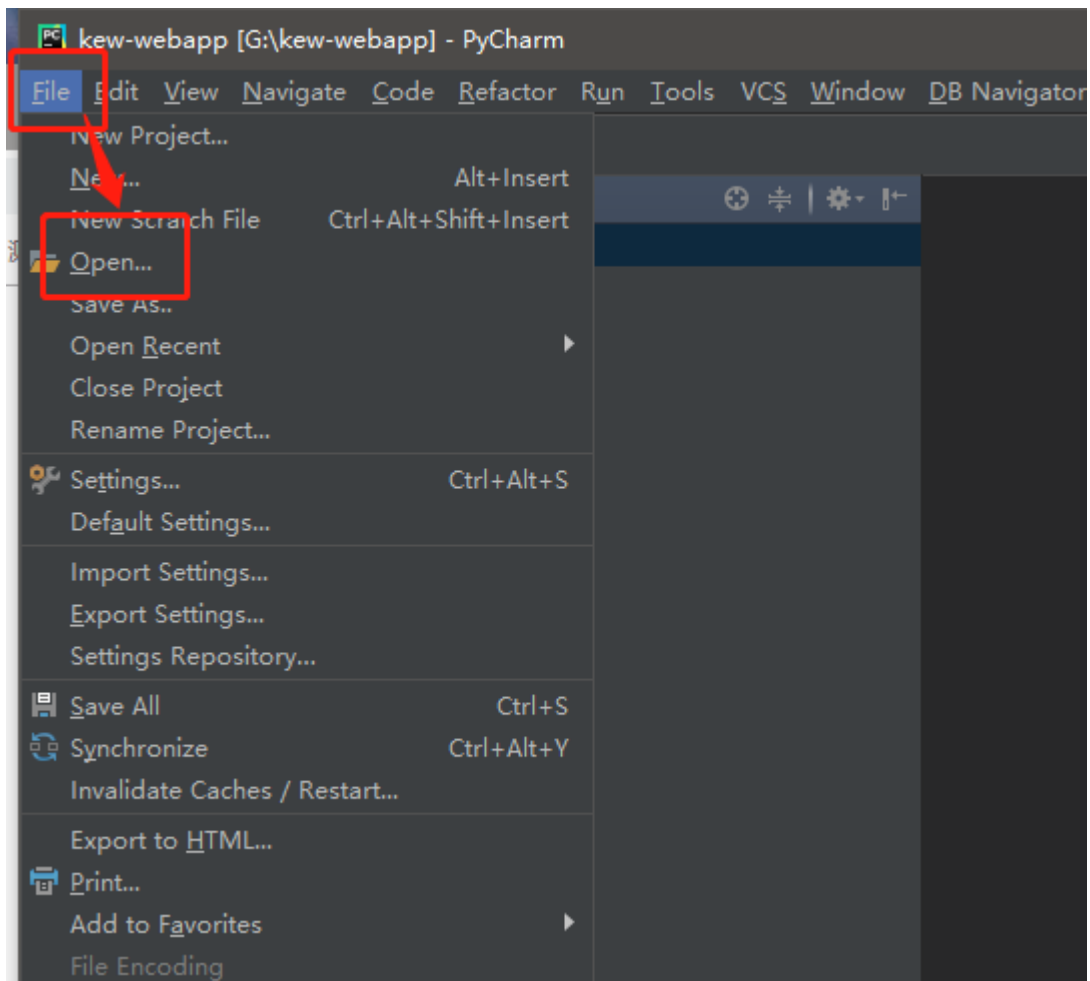
下载地址：<https://www.python.org/downloads/>



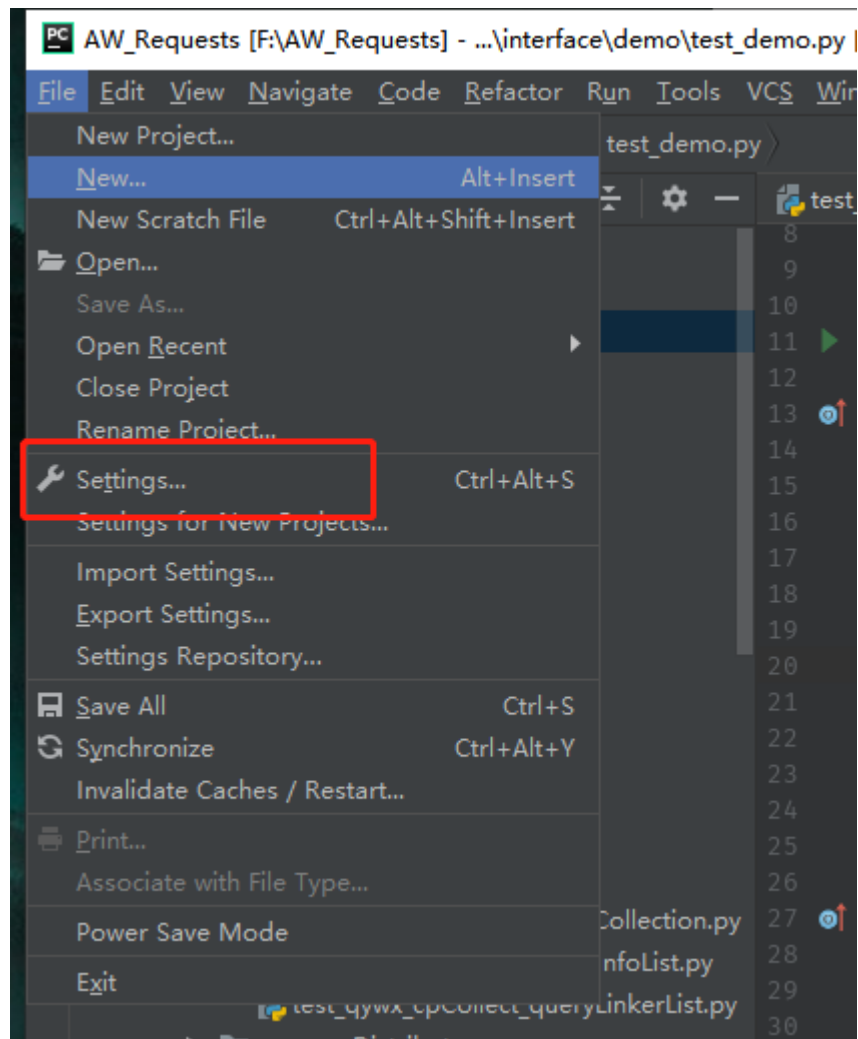
- 安装pycharm

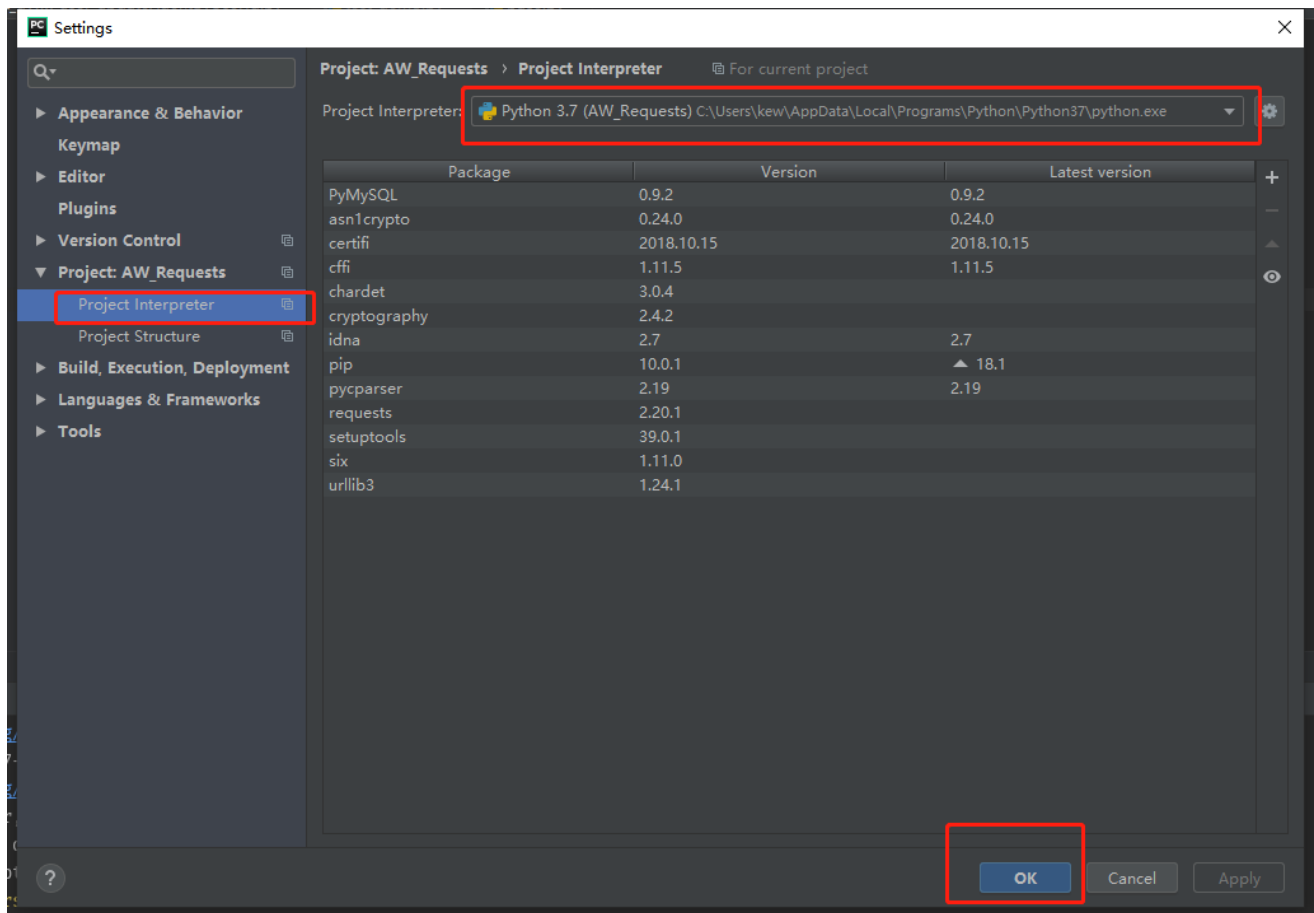
下载地址：<https://www.jetbrains.com/pycharm/>

导入AW接口框架



配置，把安装的python文件地址导入





- 安装requests

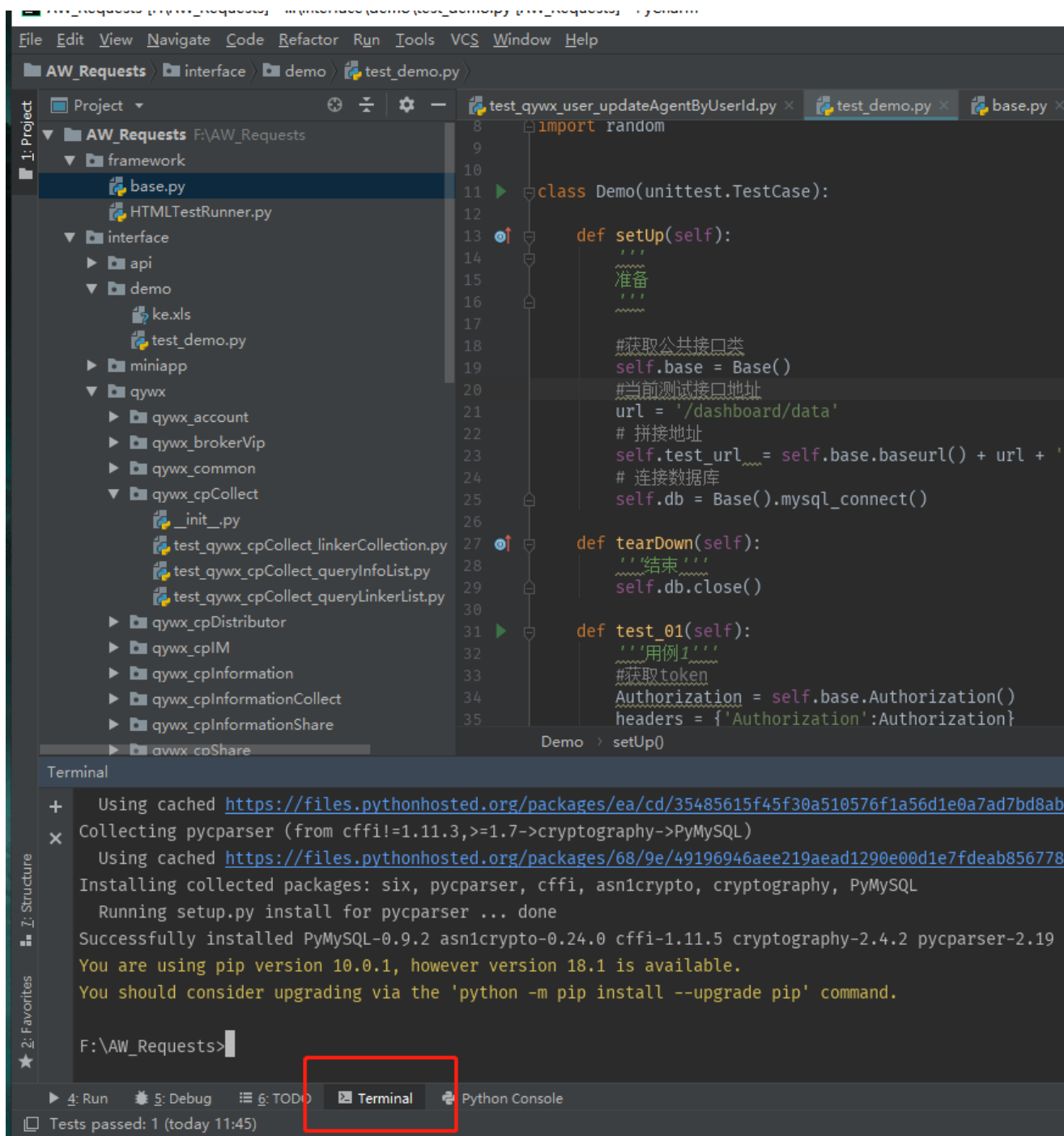
pip install requests

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



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": [
3      {
4        "classify": "user_head_img",
5        "createId": "1",
6        "createTime": "2018-08-16 22:50:10",
7        "id": "136",
8        "itemCode": "http://7201jq2test-10037467.file.myqcloud.com/ljqzs/user_head_img/male_001.png",
9        "itemName": "头像男001",
10       "sort": 2,
11       "status": 1,
12       "updateTime": "2018-08-16 22:50:10"
13     },
14     {
15       "classify": "user_head_img",
16       "createId": "1",
17       "createTime": "2018-08-16 22:50:10",
18       "id": "137",
19       "itemCode": "http://7201jq2test-10037467.file.myqcloud.com/ljqzs/user_head_img/male_002.png",
20       "itemName": "头像男002",
21       "sort": 2,
22       "status": 1,
23       "updateTime": "2018-08-16 22:50:10"
24     }
25   ],
26   { },
27   { },
28   { },
29   { },
30   { },
31   { },
32   { },
33   { },
34   { },
35   { },
36   { },
37   { },
38   { },
39   { },
40   { },
41   { },
42   { },
43   { },
44   { },
45   { },
46   { },
47   { },
48   { },
49   { },
50   { },
51   { },
52   { },
53   { },
54   { },
55   { },
56   { },
57   { },
58   { },
59   { },
60   { },
61   { },
62   { },
63   { },
64   { },
65   { },
66   { },
67   { },
68   { },
69   { },
70   { },
71   { },
72   { },
73   { },
74   { },
75   { },
76   { },
77   { },
78   { },
79   { },
80   { },
81   { },
82   { },
83   { },
84   { },
85   { },
86   { },
87   { },
88   { },
89   { },
90   { },
91   { },
92   { },
93   { },
94   { },
95   { },
96   { },
97   { },
98   { },
99   { },
100  { },
101  { },
102  { },
103  { },
104  { },
105  { },
106  { },
107  { },
108  { },
109  { },
110  { },
111  { },
112  { },
113  { },
114  { },
115  { },
116  { },
117  { },
118  { },
119  { },
120  { },
121  { },
122  { },
123  { },
124  { },
125  { },
126  { },
127  { },
128  { },
129  { },
130  { },
131  { },
132  { },
133  { },
134  { },
135  { },
136  { },
137  { },
138  { },
139  { },
140  { },
141  { },
142  { },
143  { },
144  { },
145  { },
146  { },
147  { },
148  { },
149  { },
150  { },
151  { },
152  { },
153  { },
154  { },
155  { },
156  { },
157  { },
158  { },
159  { },
160  { },
161  { },
162  { },
163  { },
164  { },
165  { },
166  { },
167  { },
168  { },
169  { },
170  { },
171  { },
172  { },
173  { },
174  { },
175  { },
176  { },
177  { },
178  { },
179  { },
180  { },
181  { },
182  { },
183  { },
184  { },
185  { },
186  { },
187  { },
188  { },
189  { },
190  { },
191  { },
192  { },
193  { },
194  { },
195  { },
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'
    labelList = "[{'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, 'labelList': labelList}

    #请求
    r = requests.put(self.test_url, data=params, headers=headers)
    #断言
    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')
```

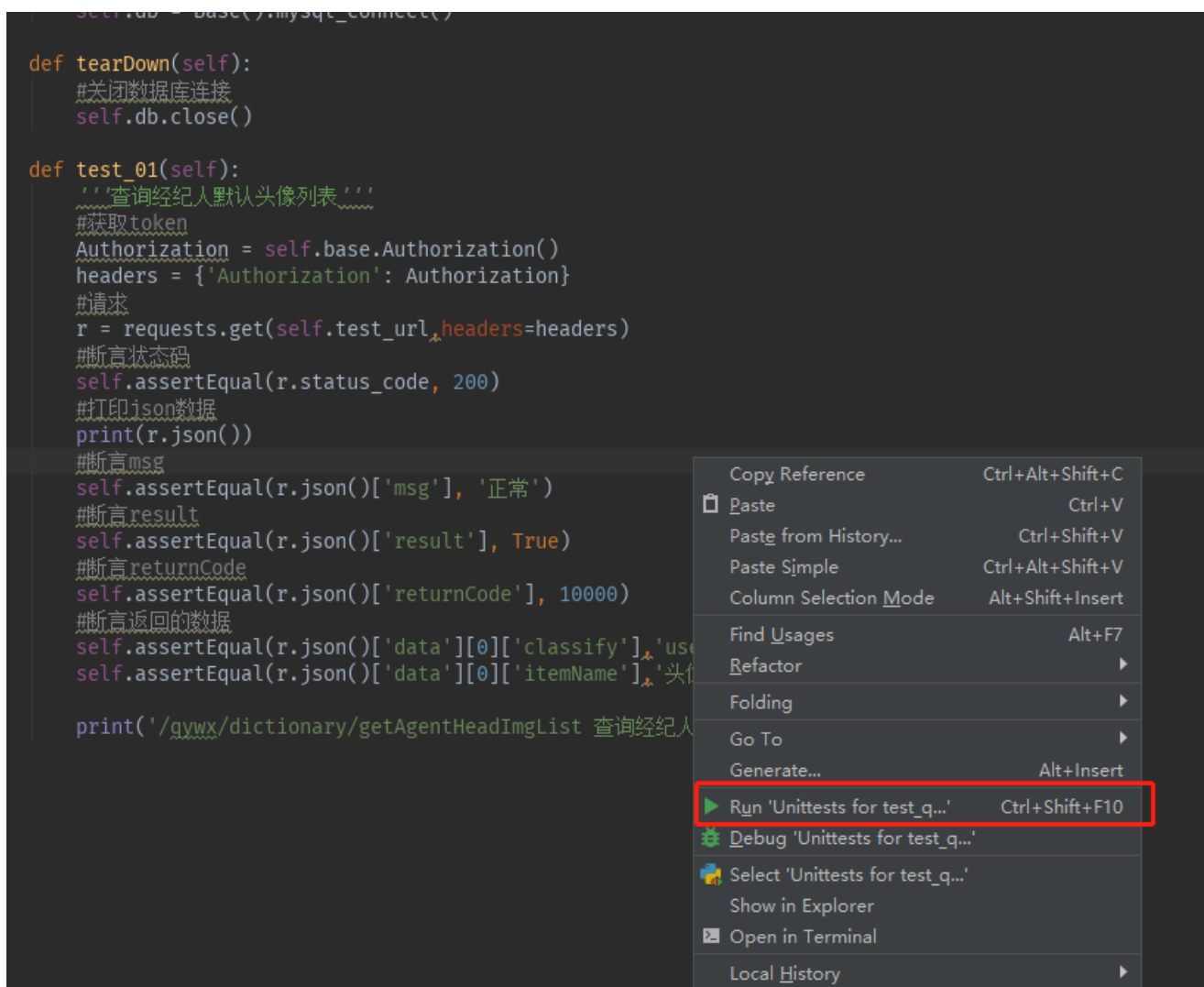
3.查询数据库信息

```
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。