

项目实战:搭建

Python+Requests+Unittest+ddt+yaml+HTMLTestRunner+mysql+selenium+邮件自动化基于数据驱动的自动化测试框架

- 1 目的:
- 2 配置数据存储到yaml中
- 3 数据存储在数据库中
- 4 把数据库中的数据进行翻译(数据库中存储的是百度首页左上角的链接文字的英文)
- 5 打开百度,依据翻译后的内容进行点击链接,然后判断是否点击了,并且正确
- 6 形成测试执行报告,然后通过邮件发送给Boss

项目准备

- 1 1.写yaml文件
- 2 百度翻译接口的url
- 3 百度翻译接口的参数
- 4 百度翻译接口的appid
- 5 百度翻译接口的密钥
- 6 百度翻译接口的随机数
- 7 百度翻译接口的语言代码
- 8 百度翻译接口的请求方式
- 9 百度首页的url
- 10
- 11
- 12 2.在数据库中创建数据库和表格及添加数据
- 13
- 14
- 15

项目规划

- 1 1.01综合案例.py:是执行单元测试类,并生成测试报告,发送邮件
- 2 2.mysqlenium.py:点击百度首页,单元测试类,并使用ddt获取yaml数据
- 3 3.mytools.py:工具模块,里面的工具有,md5加密,发送邮件,获取数据库中的数据,翻译方法

项目代码

```
1  yaml的内容:
2  appid: '20210507000816969'
3  baidu_url: https://fanyi-api.baidu.com/api/trans/vip/translate
4  data:
5  - q
6  - from
7  - to
```

```
8 - appid
9 - salt
10 - sign
11 from: en
12 miyao: sGZsjo0Y20ThznTznsRT
13 salt: '123456'
14 to: zh
15 type: get
16 url: https://www.baidu.com
17
18
19 01综合案例.py(未完成)
20 #!/usr/bin/env python
21 # -*- coding:utf-8 -*-
22 #====#====#====#====
23 #Author:
24 #CreatDate:
25 #Version:
26 #====#====#====#====
27 import unittest
28 import myselenium
29 if __name__=="__main__":
30     suit = unittest.TestSuite()
31     # 获取ddt驱动之后的测试用例名字
32     mylist = unittest.TestLoader().getTestCaseNames(myselenium.Test_Maker)
33     suit.addTests(map(myselenium.Test_Maker, mylist))
34     re = unittest.TestResult()
35     suit.run(re)
36
37 myselenium.py(未完成)
38 #!/usr/bin/env python
39 # -*- coding:utf-8 -*-
40 #====#====#====#====
41 #Author:
42 #CreatDate:
43 #Version:
44 #====#====#====#====
45 #该文件是写单元测试类
46 import unittest
47 import ddt
48 from mytools import getMysql
49 import time
50 from selenium import webdriver
51
52 @ddt.ddt
53 class Test_Maker(unittest.TestCase):
54     @classmethod
55     def setUpClass(cls):
56         cls.mylist=getMysql()
57
58     def setUp(self):
59         print("setUp")
60         print(self.mylist)
```

```

61         self.dr=webdriver.Firefox()
62
63         @ddt.file_data("conf.yaml") # 参数是文件名
64         def test_01(self,txt):
65             print(self.mylist)
66             self.dr.get(txt[-1]["url"])
67
68
69
70         def tearDown(self):
71             time.sleep(2)
72             self.dr.quit()
73
74 mytools.py
75 #!/usr/bin/env python
76 # -*- coding:utf-8 -*-
77 #====#====#====#====
78 #Author:
79 #CreateDate:
80 #Version:
81 #====#====#====#====
82 import hashlib
83 #构建邮件内容的
84 from email.mime.text import MIMEText
85 #构建邮件头部信息的
86 from email.header import Header
87 #构建发件人
88 from email.utils import formataddr
89 #创建发送邮件对象
90 import smtplib
91 import pymysql
92 import requests
93 #该文件写一些方法
94 #加密函数
95 def MyMd5(mystr):
96     mdmystr=hashlib.md5(mystr.encode(encoding='utf-8')).hexdigest()
97     return mdmystr
98
99 #发送邮件
100 def mySendEmail(filename,dis):
101     # 读取报告的内容
102     htmlmsg = open(filename,'rb').read()
103     # 构建邮件正文
104     msg = MIMEText(htmlmsg, 'html', 'utf-8')
105     # 头部信息
106     msg['Subject'] = Header("测试报告", 'utf-8')
107     # 发件人信息
108     msg['From'] = formataddr(['职员张某某202305', "76754438@qq.com"])
109     # 收件人
110     msg['To'] = dis
111
112     # 构建SMTP对象
113     smtp = smtplib.SMTP()

```

```

114     # 连接发送邮件的邮箱服务器
115     smtp.connect("smtp.qq.com")
116     # 登录
117     smtp.login("76754438@qq.com", "uomuthspmlfqbhcg")
118     # 发送
119     smtp.sendmail("76754438@qq.com", dis, msg.as_string())
120     # 退出
121     smtp.quit()
122     print("邮件发送成功...")
123
124 # 获取数据库中的数据
125 def getMysql():
126     mylist=[]
127     db = pymysql.connect(host='127.0.0.1', port=3306, user="root",
128     passwd='123456', db="mytest202305", charset='utf8')
129     # 创建游标
130     cur = db.cursor()
131     sql = "select value from myvocabulary"
132     cur.execute(sql)
133     data = cur.fetchall()
134     for i in data:
135         mylist.append(i[0])
136
137     return mylist
138
139 '''
140 [
141 {'appid': '20210507000816969'},
142 {'baidu_url': 'https://fanyi-api.baidu.com/api/trans/vip/translate'},
143 {'data': ['q', 'from', 'to', 'appid', 'salt', 'sign']},
144 {'from': 'en'},
145 {'miyao': 'sGZsjo0Y20ThznTznsRT'},
146 {'salt': '123456'},
147 {'to': 'zh'},
148 {'type': 'get'},
149 {'url': 'https://www.baidu.com'}]
150 '''
151
152 # 翻译方法
153 def myTranslate(cdata,data):
154     dis=""
155     url = cdata[1]['baidu_url']
156     mdmystr = MyMd5(cdata[0]["appid"] + data+ cdata[5]["salt"]+cdata[4]["miyao"])
157     # 参数
158     data = {
159         cdata[2]["data"][0]: data,
160         cdata[2]["data"][1]: cdata[3]["from"],
161         cdata[2]["data"][2]: cdata[6]["to"],
162         cdata[2]["data"][3]: cdata[0]["appid"],
163         cdata[2]["data"][4]: cdata[5]["salt"],
164         cdata[2]["data"][5]: mdmystr
165     }

```

```
166 # 20210507000816969test123456sGZsjo0Y20ThznTznsRT
167 # 请求
168 res = requests.get(url, params=data)
169 # 把响应转换为json格式,便于后面获取里面的内容
170 jr = res.json()
171 # 获取响应后的翻译结果
172 rdata = jr['trans_result'][0]['dst']
173 # 消息-新闻 铁巴-贴吧 相片-图片 网络磁盘-网盘
174 if rdata=="消息":
175     rdata="新闻"
176 elif rdata=="铁巴":
177     rdata="贴吧"
178 elif rdata=="相片":
179     rdata="图片"
180 elif rdata=="网络磁盘":
181     rdata="网盘"
182
183 dis=rdata
184 return dis
185
186
187
188
189
```