1.部分测试用例，检查点为：'errorCode':0，响应中也有'errorCode':0，所以运行结果显示“成功”，

但其实响应中的data字段里没有数据（数据库中没有数据导致），应该为‘失败’才对

解决方法1：

修改检查点，取data里某一个字段，这样如果检测到data里没有这个字段，就会‘失败’

响应为：{data:[{'name':'平安银行'}]}，检查点最初改为：

'name' ----带引号-------虽然响应中能找到这个数据，但这条用例仍然运行‘失败’

后来改为了：

name ----无引号--------如果响应中能找到这个数据，则这条用例运行显示‘成功'

解决方法2：

检查点由死的变为活的，查询数据库之后，把某个字段的值，插入testcase检查点这一列

2.excel中的检查点

验证接口本身

验证接口的响应数据是否和数据库匹配（excel列A为查询语句，执行完毕再写入列B，作为检查点）

验证数据库的数据是否正确（比如拿我们自己库里的数据和通达信的数据去对比）

3. 接口依赖

响应内容={'errCode': 0, 'data': [{'backtestEvaluationIndexVo': None, 'simulate': None, 'strategyName': '勿删测试用', 'userId': '27e3ba42-d6d0-4e72-9ac4-105984297fb3', 'updateDate': '2018-07-14 11:31:04', 'backtestId': '735aee9a-6f6d-4e2f-b618-66f00004e118', 'backtestCnt': 6, 'createDate': '2018-07-03 08:43:23', 'backtestStatus': 1, 'backtestStartDate': 20170702, 'backtestEndDate': 20180702, 'strategyId': 'b3602be0-fbf3-49a2-90b2-2c4d62903ffd'}], 'msg': 'success'}

要从其中取出strategyId作为后续接口的某个请求参数

关联参数的格式定义为：${strategyId}=[data][strategyId]

如果是 {'a':{'b':'C'}}的格式就好了，但现在是 {'a':[{'b':'C'}]}的格式

解决方法：

加了如下判断，首先判断是否是list，如果是的话再判断list的长度，以便取第一或第二条数据

if type(response1) is list: # 如果是列表，取第二条数据或者第一条数据

if len(type(response)) >= 2:

response1 = response1[1]

else:

response1 = response1[0]

print('response1=', response1)

correlation\_dict[param[0]] = response1 # 关联到的响应放到字典里，方便后续去遍历替换参数

4.接口依赖出现的问题

有单个关联参数，以下两种格式，程序均可以运行：

${check\_status}=[check\_status]

${check\_status}=[check\_status][cs]

有多个关联参数，如果是以下格式：

${check\_status}=[check\_status];${username}=[username];${sex}=[sex]

报错，因为：

最初的response记为A，遍历第一个关联参数之后，生成了一个response，记为B，然后再遍历第二个关联参数，用到的是B，而不是最初的原始的A，这样根据key就找不到value了

解决方法：

在第二个循环里加了一句： respon1=respon，之后都使用的是response1

for key in param[1][1:-1].split(']['):

print("key=",key)

respon1=respon

print('respon1=',respon1)

有多个关联参数，如果是以下格式：

${check\_status}=[check\_status][cs];${username}=[username];${sex}=[sex][sex1]

又会报错，因为：

对于${check\_status}=[check\_status][cs]这种格式，第一次循环没问题，第二次循环时，key为cs，response1却为原始的响应，所以根据key找不到value

最终的解决方法：在第一个循环里加该语句，respon1=respon，之后都使用的是response1

for j in correlation:

param = j.split('=') #根据=把关联数据拆分

print("param[0]=",param[0])

print("param[1]=",param[1])

respon1=respon

5.[token鉴权](https://www.cnblogs.com/tangqiu/p/7637556.html)

有些登录使用cookie，有些登录需要token验证。

token传参一般有两种形式，一种是在请求头中，一种是使用URL传参

这里举例说明一下请求头中的token方式：

#登录

param1={'username':'xxx','password':'xxxx'}

r1=requests.post('http://127.0.0.1:3000/login',data=param1)

print(r1.text)

print(r1.status\_code)

#获取登录后token

dic1=json.loads(r1.text)

token1=dic1['token']

print(token1)

#token添加到请求头中，访问

header={'Authorization':'Bearer '+token1}

r2=requests.get('http://127.0.0.1:3000/api/tasks',headers=header)

6.会话维持：session

requests.get('http://httpbin.org/cookies/set?number=123456')

res = requests.get('http://httpbin.org/cookies')

以上发了2次request请求（不管是get还是post），相当于打开了两个浏览器的窗口，他们是不相干的

s=requests.Session()

s.get('http://httpbin.org/cookies/set?number=123456')

res = s.get('http://httpbin.org/cookies')

登录一个系统之后，如果需要在登录状态下进行一些操作，那么需要怎样保持会话呢？ s=requests.Session() url1='http://127.0.0.1/wordpress/wp-login.php' data1={'log':'xxxx','pwd':'xxxxxx'} #登录 r1=s.post(url1,data=data1) print(r1.url)

#url2需要在登录状态下才能完成操作

data2={'user\_login':'test1','email':'1234567@163.com','createuser':'添加用户'} url2='http://127.0.0.1/wordpress/wp-admin/user-new.php' r2=s.post(url2,data=data2)

7.获取cookie

res=requests.get('http://www.baidu.com')

print(res.cookies)

for key, value in res.cookies.items():

print(key, value)

cookie绕过登录

使用fiddler获取登录cookie，对比登录前和登录后的cookis。

获得cookie之后（通过问开发或者抓包），使用cookie访问，就可以获取登录态：

r1=requests.get(url,cookies=cookie)

8.重定向

r = s.get('https://i.cnblogs.com/EditPosts.aspx?opt=1',headers=headers,allow\_redirects=True,verify=False)

print r.status\_code

new\_url = r.headers["Location"]

print new\_url

代理设置

proxies={'http':'http://127.0.0.1:xxxxx'}

proxies={'https':'https://127.0.0.1:xxxxx'}

proxies={'http':'http://user:password@127.0.0.1:xxxxx'}

res=requests.get('http://www.baidu.com', proxies=proxies)

超时设置

res=requests.get('http://www.baidu.com', timeout=10)

9.邮件的附件是excel表格？太沉重，后续优化，看能不能给个报告。

excel统计页，截图，是否可行？

生成测试报告，看51cto上购买过的视频

10.Excel中，设计测试用例时，有一定的逻辑在里面，按照一定的顺序，从而确保下次运行仍然能够顺利执行，比如下面：

|  |  |
| --- | --- |
| 141 | 自选股-查询是否加到默认自选股-未加入 |
| 142 | 自选股-添加自选股(加号) |
| 143 | 自选股-查询是否加到默认自选股-加入 |
| 144 | 自选股-删除自选股（减号） |

31.自动化测试在项目中的实际应用：

（4）写监控脚本，拿客户端的数据和网站的数据/手机APP的数据做对比，如果客户端的数据不是最新，则通过发送企业微信通知相关人

33.添加几个新闻不更新监控：

拿金融界网站的数据和客户端数据（来自数据库）做对比

拿手机APP的数据和客户端数据（来自数据库）做对比

部署到了jenkins，15分钟跑一次脚本，只要客户端最新数据的时间晚于网站最新数据或者ＡPP最新数据，就发微信给数据人员（可能会有误报，但是如果连续的报错，就能看出来客户端的新闻数据是不是异常了，是不是一直在更新，还是停止了更新）

使用以下语句，以免从网站获取到的标题显示乱码

res = requests.get(url).content.decode('gbk', 'ignore').encode('utf8', 'ignore')

50.python查询mongo数据库：

find\_one()， 查询一条记录， 以字典的形式直接返回数据结果

find()， 查询多条记录， 返回的是Cursor对象，需要遍历才能获取数据

# 查询单条记录 results = collection.find\_one({'name': '五年高考'}) print(result) # 查询多条记录 results = collection.find({'price': 50}) for result in results: print(result)

查询不到则返回None

53.CentOS6.5环境搭建，安装python3.6（自动安装pip3），然后把程序部署到CentOS上，并设置定时任务(crontab)

（1）需要改为绝对路径

testcase= '/fcy/mypyproject\_linux/testcase/testcase.xlsx'

（2）需要在包含中文的py文件里添加

# encoding:utf-8

（3）如果from XX import XX提示没有找到XX module（XX为自己定义的目录）：

可以考虑把所有py文件都放在一个文件夹下，然后删掉from XX即可

（4）使用pip3 install XXX ， 以及 pip3 list查看

55.python查询数据库，会用到sort排序字段

for data1 in collection.find().sort(sortfield).limit(1):

sortfield，在excel中的格式不是这样：({'trade\_date':-1, 'trade\_min':-1}) ， 而是以下写法：

|  |
| --- |
| 排序字段 |
| [('trade\_date',-1),('trade\_min',-1)] |

58.此项目已经可以在linux上运行，并能够根据设置的定时任务crontab定时执行。需要解决的问题：

网络的问题（如何使用SSH远程连接到这个机器）

VPN的问题（如果是运维登记过的服务器，就不存在这个问题了）

64.爬虫

存储到文本文件、Excel、mongodb

利用正则、beautifulsoup、pyquery分别实现

串行

并行---多线程

并行---多进程

65.学会了2点：

（1）利用openpyxl存储数据到excel，尤其是利用table.append(列表/元组)

（2）把爬取出来的所有数据放到一个列表里或者字典里，使用return返回，供后续使用

66.requests请求网址，读取文件内容，然后再写入到本地某路径的文件中（文件下载）

写入txt

r = requests.get(url, headers=headers, verify=False)

file = r.text

with open('aaa.txt', 'w', encoding='utf-8') as f:

f.write(file)

写入excel

r = requests.get(url, headers=headers, verify=False)

file = r.content

with open('aaa.xlsx', 'wb') as f:

f.write(file)

67.bytes转化为str类型，需要encoding才可以转换成功，指定encoding才可以显示中文

res = res.content

res = str(res, encoding='gbk')

68.注意里面这个判断的写法

info = {

'startTime':data['startTime'],

'id':data['id'],

'cityName':data['cityName'] if 'cityName' in data else '',

'score':data['score'],

'nickName':data['nickName'],

'content':data['content'].replace('\n',' ') #处理换行

}

75. hasattr getattr的用法

79. 搜索：python练习题

以下就是路人甲精心给大家整理的Python练手项目：

1、show-me-your-code：https://github.com/Yixiaohan/show-me-the-code

2、Python练手100例：<http://www.runoob.com/python/python-100-examples.html>