requests库课件

requests库是一个常用的用于http请求的模块，它使用python语言编写，在当下python系列的接口自动化中应用广泛，本文将带领大家深入学习这个库

Python环境的安装就不在这里赘述了，我们直接开干

一. requests的安装

windows下执行如下命令

```
pip install requests -i http://pypi.douban.com/simple/ --trust-host
pypi.douban.com
```

mac终端下执行如下命令

```
python3 -m pip install requests -i http://pypi.douban.com/simple/ --trust-host
pypi.douban.com
```

二. 自动化requests模块的常用方法

方法	含义
requests.get()	发起get请求调用
requests.post()	发起post请求调用
requests.put()	发起put请求调用
requests.delete()	发起delete请求调用
requests.session()	获取requests的session对象
requests.session().request()	也是发起请求，可以自动管理cookie

2.1 get请求实战

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2021/3/25 9:54
# @Copyright：北京码同学网络科技有限公司

import requests

host = 'http://10.0.0.18:8080'
def get():
    """
    get接口请求
    :return:
    """
    url = host+'/pinter/com/getSku' #接口地址
    params = {
```

```
        'id': 1
    }
    resp = requests.get(url,params=params)
    status_code = resp.status_code #获取响应状态码
    print('响应状态码:{}'.format(status_code))
    text = resp.text #获取响应内容，结果类型是字符串
    print('响应内容:{}'.format(text))
    json = resp.json() #获取响应内容，结果是字典类型
    print('响应内容:{}'.format(json))
    resp_headers = resp.headers #获取响应headers
    print('响应header:{}'.format(resp_headers))

if __name__ == '__main__':
    get()
```

结果如下

```
D:\Python\Python36\python.exe D:/pycharmprojects/first/requetsstudy/pinter.py
响应状态码:200
响应内容:{"code":"0","message":"success","data":{"skuId":1,"skuName":"ptest-
1","price":"645","stock":709,"brand":"testfan"}}
<class 'dict'>
响应内容:{'code': '0', 'message': 'success', 'data': {'skuId': 1, 'skuName':
'ptest-1', 'price': '645', 'stock': 709, 'brand': 'testfan'}}
响应header:{'Content-Type': 'application/json;charset=UTF-8', 'Transfer-
Encoding': 'chunked', 'Date': 'Fri, 12 Mar 2021 22:13:49 GMT', 'Keep-Alive':
'timeout=20', 'Connection': 'keep-alive'}

Process finished with exit code 0
```

上述代码中请求发起后得到一个响应对象变量resp，那么resp对象的常用方法如下

方法	含义
resp.status_code	获取响应状态码
resp.text	获取响应内容，结果类型是字符串
resp.json()	获取响应内容，结果是字典类型
resp.headers	获取响应headers

2.2 post请求实战

post请求的参数格式通常有多种，我们依次学习

第一种表单形式的参数

```
import requests

host = 'http://10.0.0.18:8080'

def post():
    """
    post 表单
    :return:
```

```
    """
    url = host +'/pinter/com/login'
    #表单参数
    data = {
        'userName':'沙陌',
        'password':'123456'
    }
    resp = requests.post(url=url,data=data)
    status_code = resp.status_code #获取响应状态码
    print('响应状态码:{}'.format(status_code))
    text = resp.text #获取响应内容，结果类型是字符串
    print('响应内容:{}'.format(text))
    json = resp.json() #获取响应内容，结果是字典类型
    print('响应内容:{}'.format(json))
    resp_headers = resp.headers #获取响应headers
    print('响应header:{}'.format(resp_headers))
```

第二种json格式参数

```
import requests

host = 'http://10.0.0.18:8080'
def post_json():
    """
    post json
    :return:
    """
    url = host +'/pinter/com/register'
    # header里定义参数类型
    headers = {
        'Content-Type':'application/json'
    }
    #json参数
    json = {
         "userName": "沙陌",
         "password": "1234",
         "gender": 1,
         "phoneNum": "110",
         "email": "beihe@163.com",
         "address": "Beijing"
    }
    resp = requests.post(url=url,json=json)
    status_code = resp.status_code #获取响应状态码
    print('响应状态码:{}'.format(status_code))
    text = resp.text #获取响应内容，结果类型是字符串
    print('响应内容:{}'.format(text))
    json = resp.json() #获取响应内容，结果是字典类型
    print('响应内容:{}'.format(json))
    resp_headers = resp.headers #获取响应headers
    print('响应header:{}'.format(resp_headers))
```

2.3 put接口实战

```
import requests
host = 'http://10.0.0.18:8080'
def put():
    """
    put 清酒
    :return:
    """
    url = host + '/pinter/com/phone' #接口地址
    #参数
    json ={
        "brand":"Huawei",
        "color":"yellow",
        "memorySize":"64G",
        "cpuCore":"8核",
        "price":"8848",
        "desc":"全新上市"
    }
    resp = requests.put(url=url,json=json)
    status_code = resp.status_code #获取响应状态码
    print('响应状态码:{}'.format(status_code))
    text = resp.text #获取响应内容，结果类型是字符串
    print('响应内容:{}'.format(text))
    json = resp.json() #获取响应内容，结果是字典类型
    print('响应内容:{}'.format(json))
    resp_headers = resp.headers #获取响应headers
    print('响应header:{}'.format(resp_headers))
```

2.4 delete请求

```
import requests
host = 'http://10.0.0.18:8080'
def delete():
    """
    delete接口请求
    :return:
    """
    url = host+'/pinter/com/phone' #接口地址
    #参数
    json ={
        "brand":"Huawei",
        "color":"yellow",
        "memorySize":"64G",
        "cpuCore":"8核",
        "price":"8848",
        "desc":"全新上市"
    }
    resp = requests.delete(url=url,json=json)
    status_code = resp.status_code #获取响应状态码
    print('响应状态码:{}'.format(status_code))
    text = resp.text #获取响应内容，结果类型是字符串
    print('响应内容:{}'.format(text))
    json = resp.json() #获取响应内容，结果是字典类型
    print('响应内容:{}'.format(json))
    resp_headers = resp.headers #获取响应headers
    print('响应header:{}'.format(resp_headers))
```

2.5 request.session.request用法

可以自动管理cookie，比如如下需要采用cookie认证的接口

接口名称	接口类型	接口地址	接口参数
银行登录接口（cookie）	POST	/pinter/bank/api/login	userName=admin&password=1234
银行余额查询接口（cookie）	GET	/pinter/bank/api/query	userName=admin

```
import requests
host = 'http://10.0.0.18:8080'
session = requests.session()
def query():
    """
    查询余额
    :return:
    """
    url = host + '/pinter/bank/api/query'  # 接口地址
    params = {
        'userName': 'admin'
    }
    resp = session.request(url=url,method='get',params=params)
    status_code = resp.status_code #获取响应状态码
    print('响应状态码:{}'.format(status_code))
    text = resp.text #获取响应内容，结果类型是字符串
    print('响应内容:{}'.format(text))
def login():
    """
    登录
    :return:
    """
    url = host +'/pinter/bank/api/login'
    #表单参数
    data = {
        'userName':'admin',
        'password':'1234'
    }
    session.request(url=url,method='post',data=data)
if __name__ == '__main__':
    login()
    query()
```

结果如下：

```
D:\Python\Python36\python.exe D:/pycharmprojects/first/requetsstudy/pinter.py
响应状态码:200
响应内容:{"code":"0","message":"success","data":"$ 22,378,198"}

Process finished with exit code 0
```

2.6 token关联的接口如何做呢?

接口名称	接口类型	接口地址	接口参数
银行登录接口（token）	POST	/pinter/bank/api/login2	userName=admin&password=1234
银行余额查询接口（token）	GET	/pinter/bank/api/query2	userName=admin

对于需要token关联的接口来说，需要从登录接口的返回值中提取token信息，并传递给需要token的接口

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2021/3/25 9:54
# @Copyright：北京码同学网络科技有限公司
import jsonpath
import requests
host = 'http://10.0.0.18:8080'
session = requests.session()
def query():
    """
    查询余额
    :return:
    """
    url = host + '/pinter/bank/api/query2'  # 接口地址
    headers = {
        'testfan-token':token #加入token字段
    }
    params = {
        'userName': 'admin'
    }

    resp = session.request(url=url,method='get',params=params,headers=headers)
    status_code = resp.status_code #获取响应状态码
    print('响应状态码:{}'.format(status_code))
    text = resp.text #获取响应内容，结果类型是字符串
    print('响应内容:{}'.format(text))
def login():
    """
    登录
    :return:
    """
    global token
    url = host +'/pinter/bank/api/login2'
    #表单参数
    data = {
        'userName':'admin',
        'password':'1234'
    }
    resp = session.request(url=url,method='post',data=data)
    resp_json = resp.json()
```

```
    token = jsonpath.jsonpath(resp_json,'$.data')[0] #使用jsonpath从响应结果中提取data字段

if __name__ == '__main__':
    login()
    query()
```

结果如下：

```
D:\Python\Python36\python.exe D:/pycharmprojects/first/requetsstudy/pinter1.py
响应状态码:200
响应内容:{"code":"0","message":"success","data":"$ 74,780,457"}

Process finished with exit code 0
```

总结一下：

requests库的请求方法里参数众多，所以简单划分一下，

查询参数就用params=params

表单参数就用data=data

json参数就用json=json

请求头信息header就用headers=headers