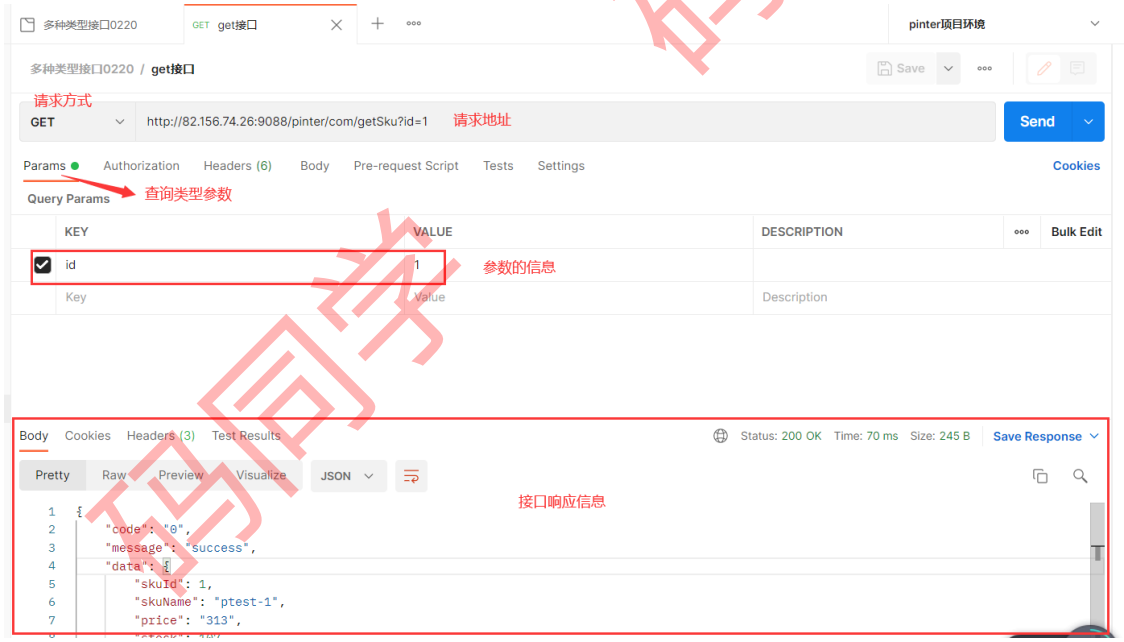
0220课堂笔记

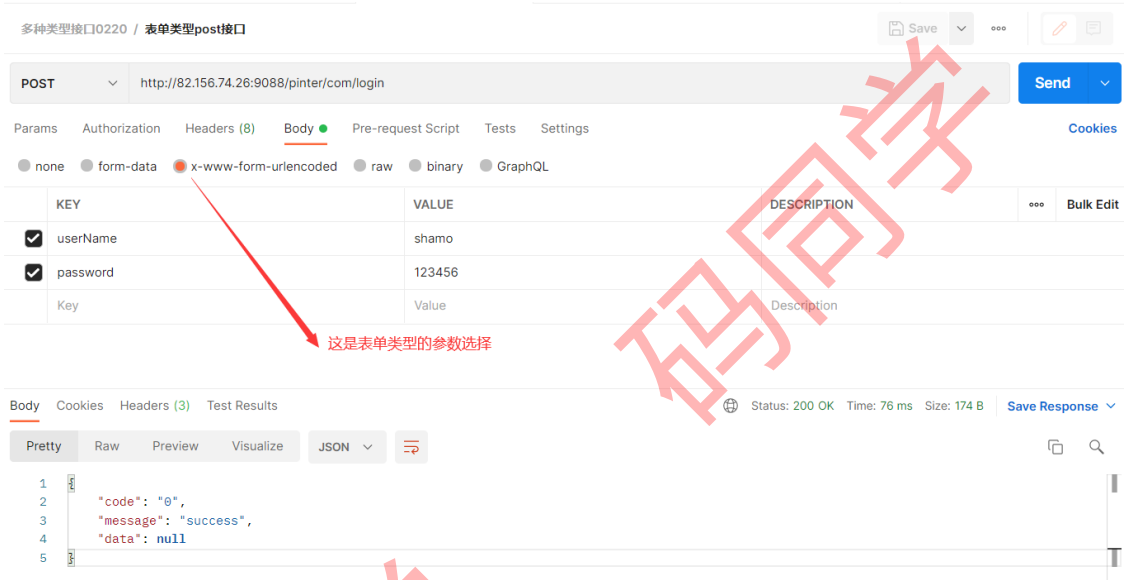
沙陌 微信：Matongxue_2

1.postman使用

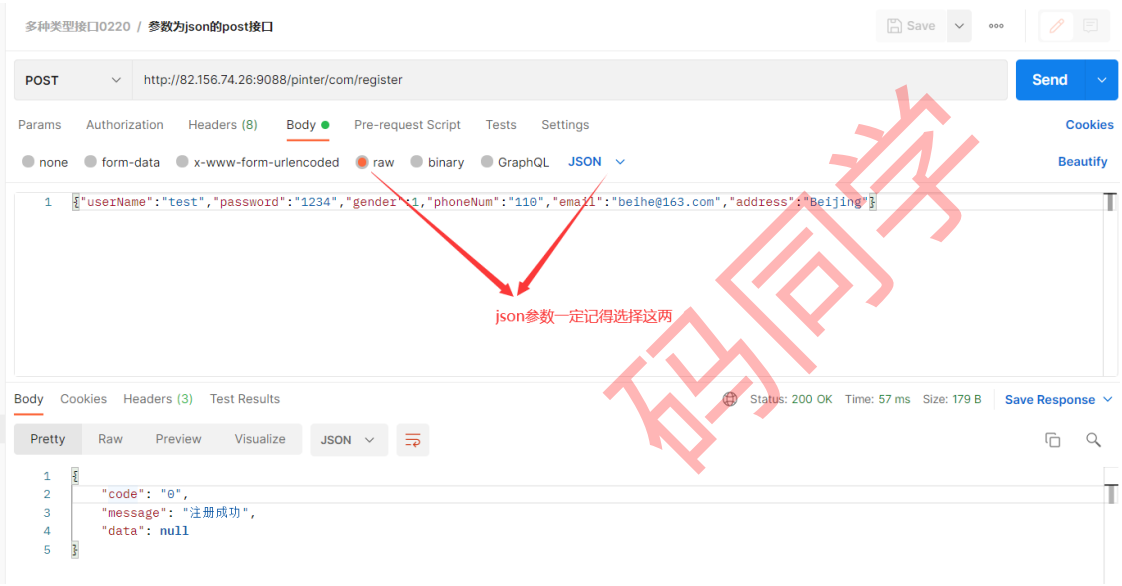
1. get接口



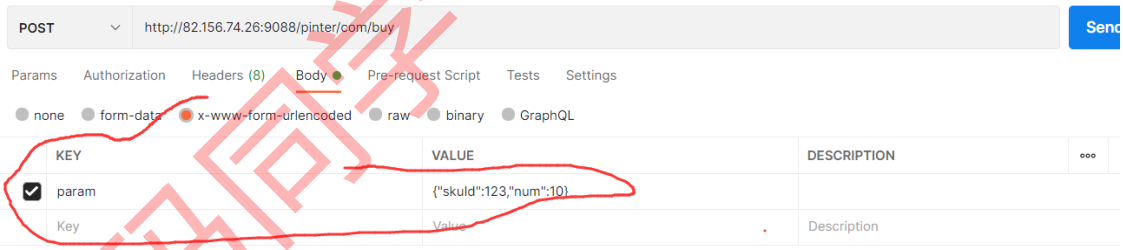
2. post表单参数接口



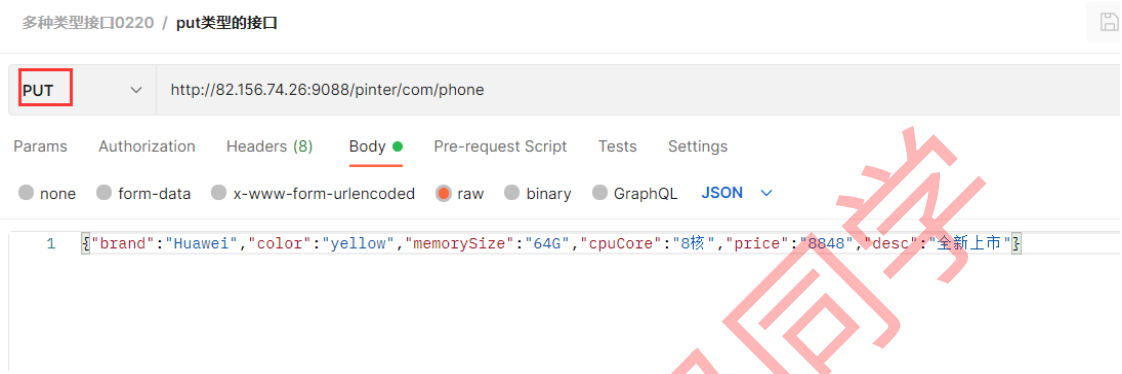
3. post json参数



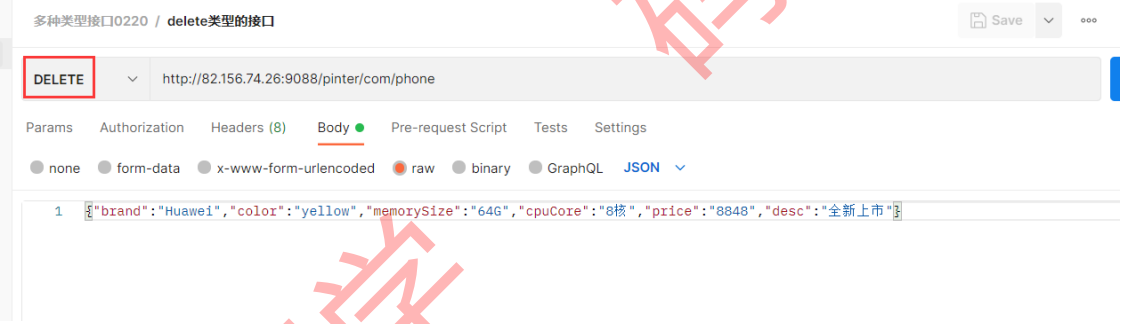
4. 参数k=json的post接口



5. put类型的接口



6. delete类型的接口

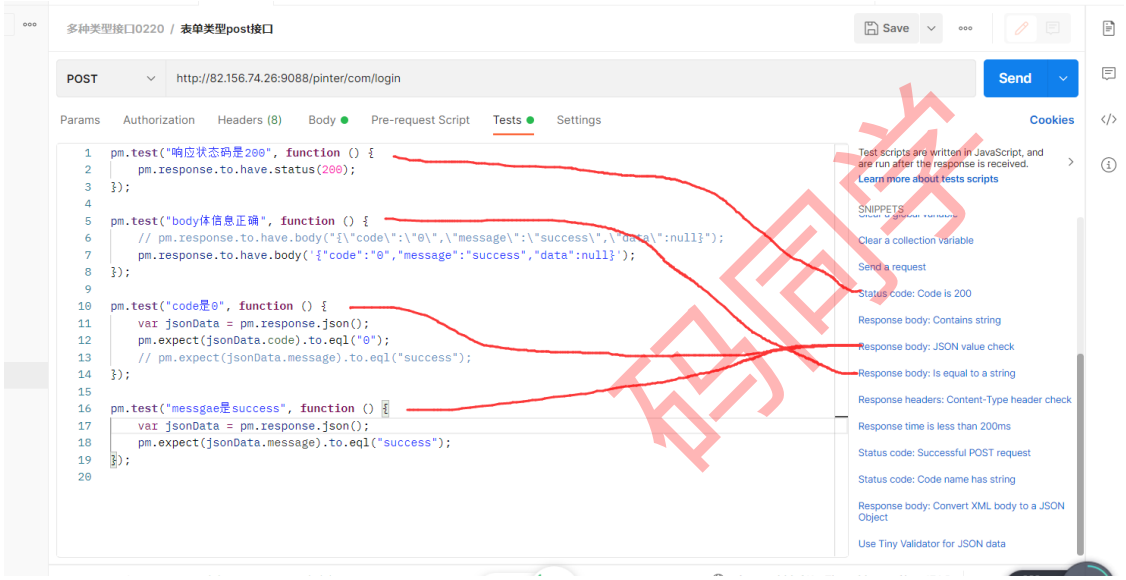


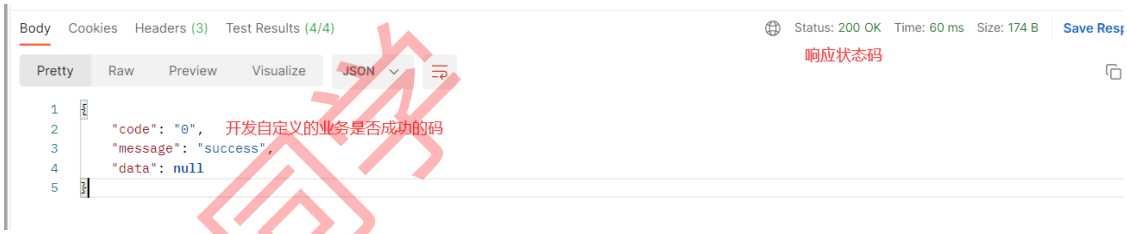
7. 断言

在之前的接口里，我们只是调通了接口，但是并没有针对接口设置相关的断言

如何判断一个接口在测试时是否符合预期，有哪些维度?

响应状态码、响应body信息、验证数据(数据库/redis/mq等等)



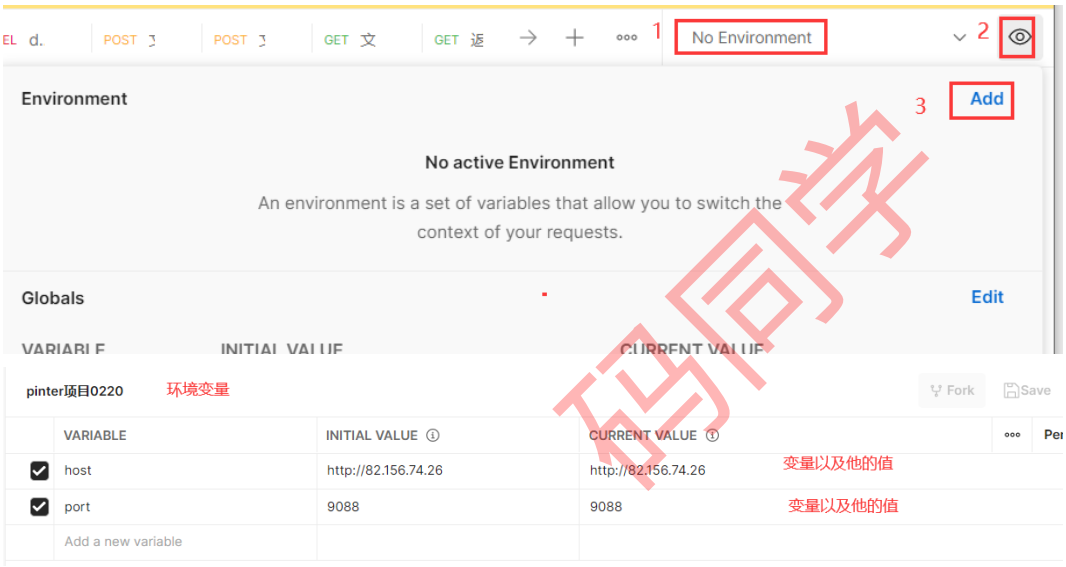


响应状态码只是代表了你的请求被处理成功了，但是业务是否成功不一定

8. 变量存储

   在接口测试脚本中，通常有一些数据是公共型的，我们可以将其存储在变量中，以便维护和调用

   - 创建环境变量



   - 引用变量

     变量调用固定格式{{xxxx}} xxxx就表示的是变量名称

9. 需要签名验证的接口

```
{
    "phoneNum":"123434",
    "optCode":"testfan",
    "timestamp":"12112121212",
    "sign":"your sign data"
}
```

签名规则是：sign字段的值=MD5(phoneNum的值+optcode的值+timestamp的值)

签名的作用有两个：

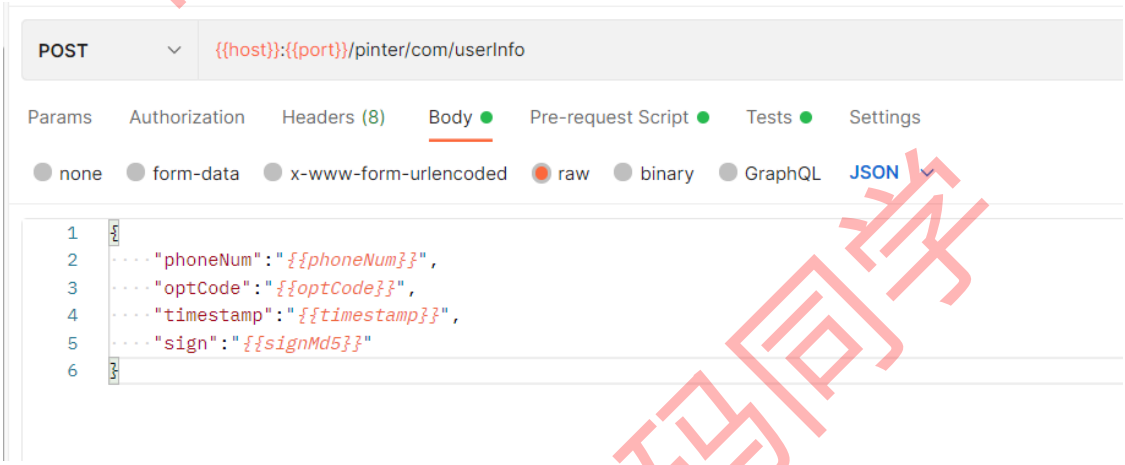
1. 防止传递的参数数据被篡改
2. 确保发送请求方是有权限的

当前这个接口是第一种

将phoneNum和optCode设置为环境变量

再完成以下加密处理

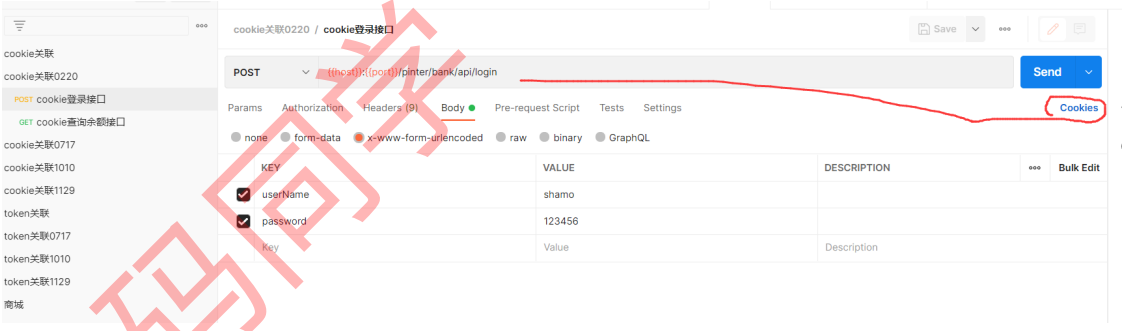
在接口参数中引用相关的变量



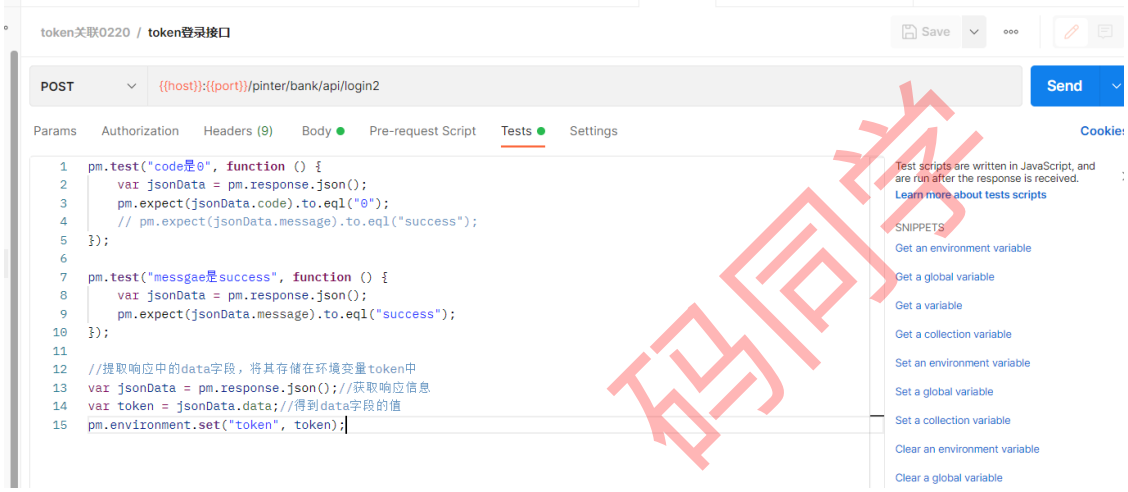
不同的项目不同的接口，签名的方式可能都不一样，所以需要找开发去问

10. cookie关联的接口

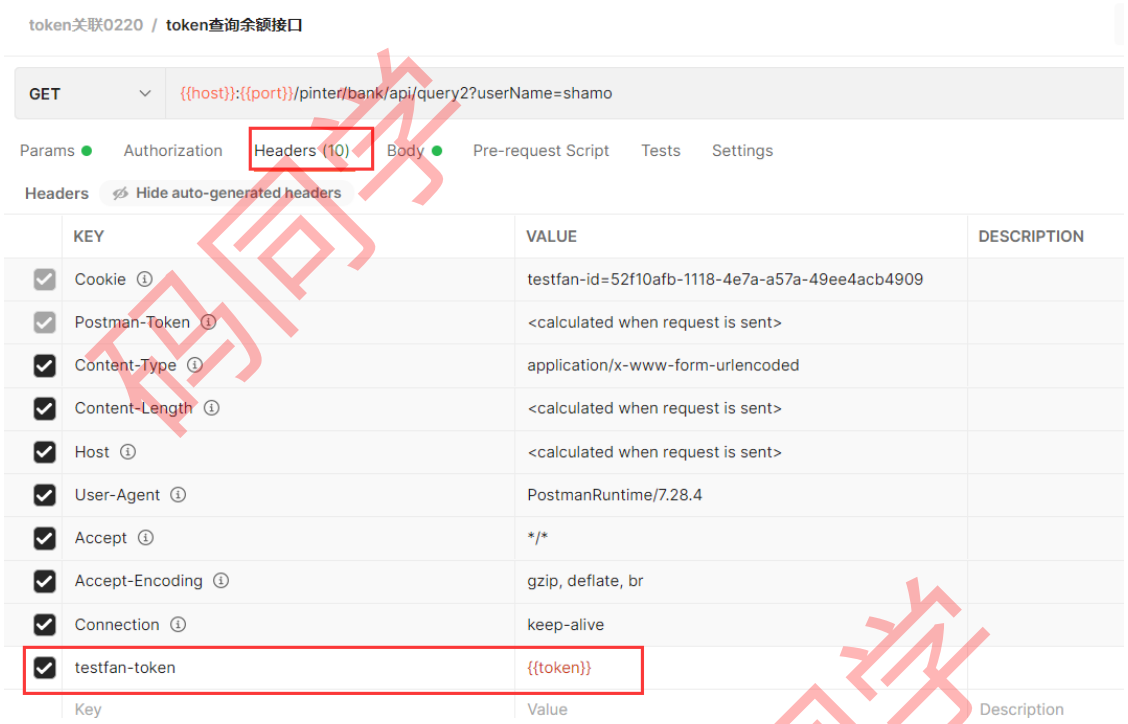
    在postman，cookie关联的接口，不需要做特殊的处理，只要在需要cookie的接口之前去执行产生cookie的接口即可，postman会自动将产生的cookie存储，之后接口调用时会带着



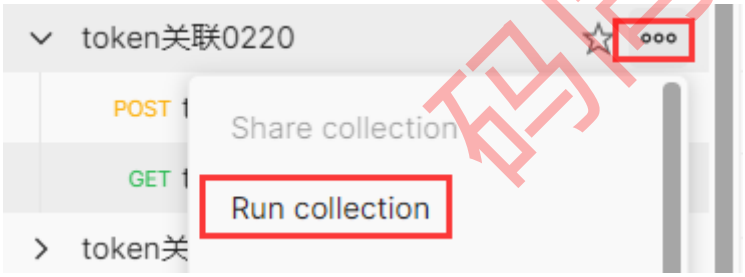
11. token关联

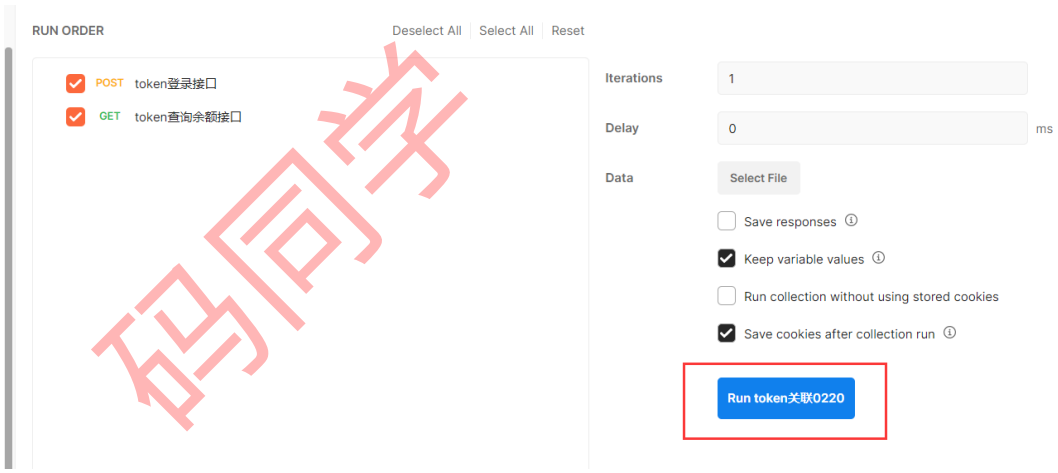


需要用到的接口中调用



执行时以测试集合运行的方式执行





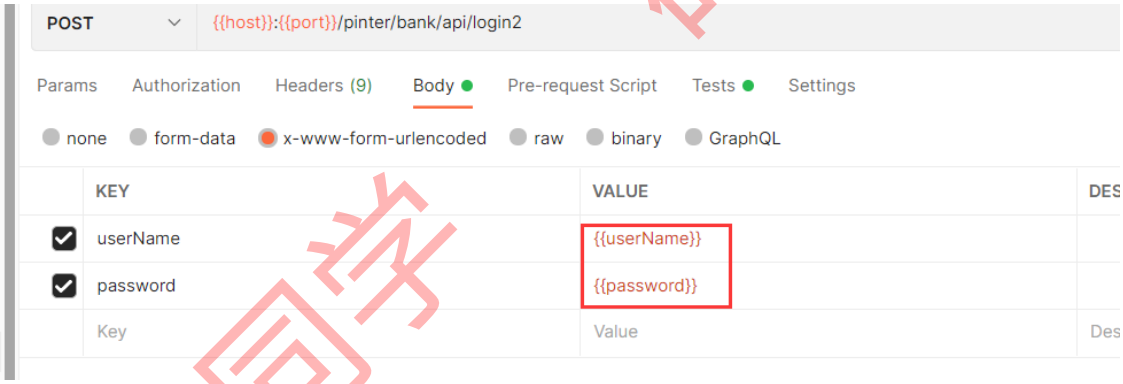
12. 数据参数化

以token登录接口为例，将登录的用户名和密码作为参数化数据，创建一个txt文件，写上如下内容

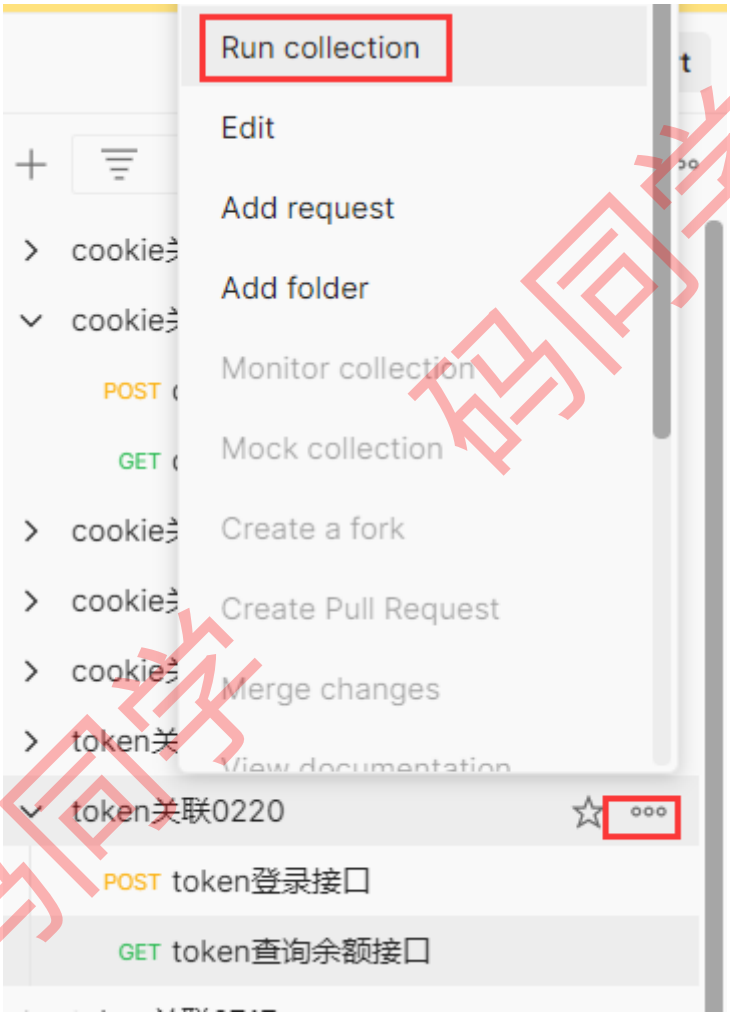
```
userName,password
shamo1,123456
shamo2,567766
shamo3,787866
```

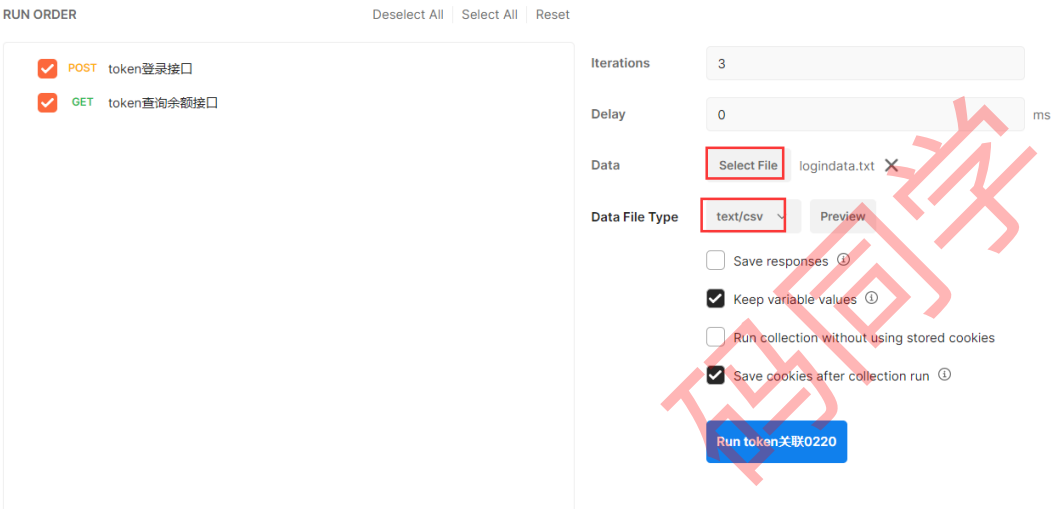
修改用到参数化数据的接口，引用相关变量

修改登录参数，以及查询余额的接口参数



以测试集合的运行方式来运行，并且选择参数化数据文件





然后点击run

2.接口测试简单总结

- 请求地址
- 请求方式：get/post/put/delete
- 请求headers：并不是必须，除非接口有特殊的信息，比如token信息通常放在headers里
- 请求参数：查看类型参数、表单参数、json参数、文件参数等等
- 响应信息

  上述的信息要么从接口文档获取，要么自己抓包获取，要么问开发获取

3.postman命令行运行

1. 安装nodejs

   https://nodejs.org/download/release/v16.0.0/

Index of /download/release/v16.0.0/

../		
docs/	20-Apr-2021 11:38	-
win-x64/	20-Apr-2021 11:17	-
win-x86/	20-Apr-2021 11:13	-
SHASUMS256.txt	20-Apr-2021 15:57	3127
SHASUMS256.txt.asc	20-Apr-2021 15:57	3664
SHASUMS256.txt.sig	20-Apr-2021 15:57	310
node-v16.0.0-aix-ppc64.tar.gz	20-Apr-2021 10:58	44489711
node-v16.0.0-darwin-arm64.tar.gz	20-Apr-2021 11:19	29368427
node-v16.0.0-darwin-arm64.tar.xz	20-Apr-2021 11:19	19028900
node-v16.0.0-darwin-x64.tar.gz	20-Apr-2021 10:56	30803023
node-v16.0.0-darwin-x64.tar.xz	20-Apr-2021 10:58	20650712
node-v16.0.0-headers.tar.gz	20-Apr-2021 11:38	551878
node-v16.0.0-headers.tar.xz	20-Apr-2021 11:38	374316
node-v16.0.0-linux-arm64.tar.gz	20-Apr-2021 10:51	33287976
node-v16.0.0-linux-arm64.tar.xz	20-Apr-2021 10:53	21595452
node-v16.0.0-linux-armv7l.tar.gz	20-Apr-2021 10:52	30607660
node-v16.0.0-linux-armv7l.tar.xz	20-Apr-2021 10:53	18621956
node-v16.0.0-linux-ppc64le.tar.gz	20-Apr-2021 10:51	35271788
node-v16.0.0-linux-ppc64le.tar.xz	20-Apr-2021 10:52	22668176
node-v16.0.0-linux-s390x.tar.gz	20-Apr-2021 10:50	33521239
node-v16.0.0-linux-s390x.tar.xz	20-Apr-2021 10:51	21150284
node-v16.0.0-linux-x64.tar.gz	20-Apr-2021 11:41	33283633
node-v16.0.0-linux-x64.tar.xz	20-Apr-2021 11:42	22206588
node-v16.0.0-win-x64.7z	20-Apr-2021 11:18	17329643
node-v16.0.0-win-x64.zip	20-Apr-2021 11:18	26841308
node-v16.0.0-win-x86.7z	20-Apr-2021 11:13	16110938
node-v16.0.0-win-x86.zip	20-Apr-2021 11:13	25092862
node-v16.0.0-x64.msi	20-Apr-2021 11:18	29138944
node-v16.0.0-x86.msi	20-Apr-2021 11:13	27283456
node-v16.0.0.pkg	20-Apr-2021 11:53	56428721
node-v16.0.0.tar.gz	20-Apr-2021 11:31	64007010
node-v16.0.0.tar.xz	20-Apr-2021 11:35	33784448

下载以后一路下一步安装即可，一般情况下会自动配置到环境变量

2. 安装cnpm

   cnpm，指的是npm的淘宝镜像命令

   npm是nodejs的项目管理工具

   npm install -g cnpm --registry=https://registry.npm.taobao.org

3. 安装newman

   newman这个工具就是用来执行postman脚本的，在命令行中

   cnpm install newman --global

4. 安装reporter

   reporter这个工具是newman在执行时用来生成html报告的

   cnpm install -g newman-reporter-html

5. 导出脚本需要的东西

   创建一个文件夹，专门用来存储postman导出的东西以及数据参数文件

6. 执行命令

   命令行中先进入到脚本文件所在的目录，再执行以下命令

```
newman run 脚本 -d 参数文件 -e 环境文件 -n 循环次数 -r html --reporter-html-export 报告路径
```

课上命令：

```
newman run token关联0220.postman_collection.json -d logindata.txt -e pinter项目0220.postman_environment.json -n 3 -r html --reporter-html-export .\
```

执行之后在目录中生成测试报告文件

4.charles

参考课件

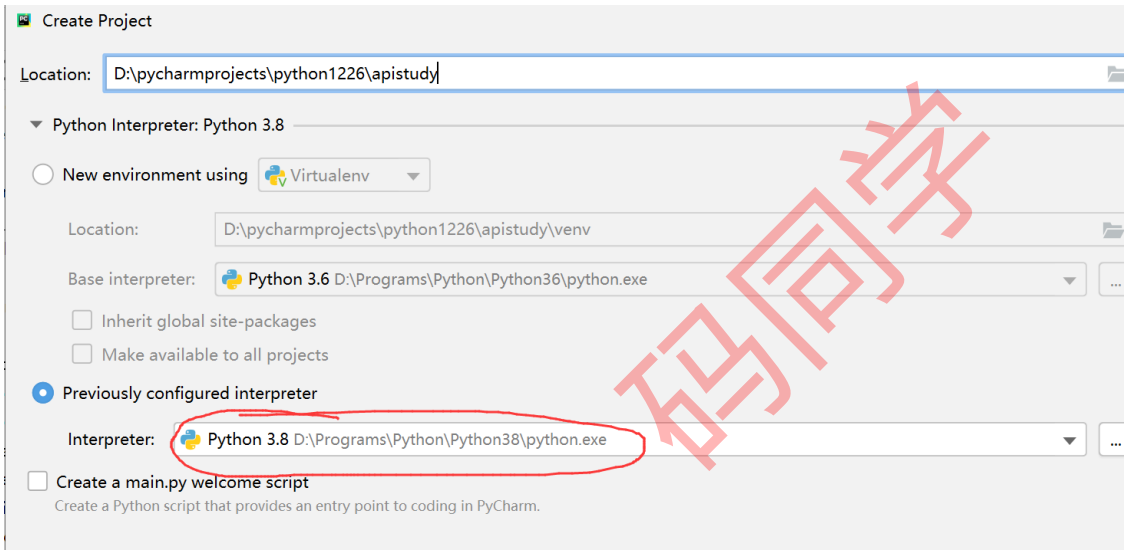
5.requests库

所谓的第三库指的是不是python自带的，但是我们要用他，需要先安装这个第三库

在命令行中执行如下命令

```
# windows
pip install requests -i https://pypi.douban.com/simple/
# mac
python3 -m pip install requests -i https://pypi.douban.com/simple/
```

- 创建项目



- get接口调用

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信: Matongxue_2
# @Time: 2022-02-20 16:24
# @Copyright: 北京码同学
import requests
# 对于一个接口的调用或者说测试，都需要知道哪些东西
# 请求地址、请求headers、请求方式、请求参数、响应信息
# 卖家登录接口
url = 'http://www.mtxshop.com:7003/seller/login'
```

```
headers = {
    'Authorization':''
}
#查询参数通常使用params来表示
params = {
    'username':'shamoseller',
    'password':'e10adc3949ba59abbe56e057f20f883e',
    'captcha':'1512',
    'uuid':'jsjdhdhdhdhdhdhh'
}

# 接口的基本信息我们已经准备好了，如何发起调用
# 先导入requests库
# 发起调用并得到响应对象
# 注意这个响应包含了响应里的所有东西，我们经常关注响应状态码、响应body信息
resp = requests.get(url=url,headers=headers,params=params)

status_code = resp.status_code #获取响应状态码
print(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)
```

- post接口

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2022-02-20 16:38
# @Copyright：北京码同学
import requests
# 对于一个接口的调用或者说测试，都需要知道哪些东西
# 请求地址、请求headers、请求方式、请求参数、响应信息
# 卖家登录接口
url = 'http://www.mtxshop.com:7002/passport/login'

headers = {
    'Authorization':''
}
#查询参数通常使用params来表示
params = {
    'username':'shamo',
    'password':'e10adc3949ba59abbe56e057f20f883e',
    'captcha':'1512',
    'uuid':'jsjdhdhdhdhdhdhh'
}
# 通常requests库针对表单参数时，使用data来表示
# data = {
#     'username':'shamo',
#     'password':'e10adc3949ba59abbe56e057f20f883e',
#     'captcha':'1512',
#     'uuid':'jsjdhdhdhdhdhdhh'
# }
# resp = requests.post(url=url,headers=headers,data=data)
```

```
# 接口的基本信息我们已经准备好了，如何发起调用
# 先导入requests库
# 发起调用并得到响应对象
# 注意这个响应包含了响应里的所有东西，我们经常关注响应状态码、响应body信息
resp = requests.post(url=url,headers=headers,params=params)

status_code = resp.status_code #获取响应状态码
print(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)


# post接口传json参数
url = 'http://82.156.74.26:9088/pinter/com/register'
# json格式的参数通常以json来表示
json =
{"userName":"test","password":"1234","gender":1,"phoneNum":"110","email":"be
ihe@163.com","address":"Beijing"}

resp = requests.post(url=url,json=json)
status_code = resp.status_code #获取响应状态码
print(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)
```

- put接口

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2022-02-20 16:48
# @Copyright：北京码同学



import requests

url = 'http://82.156.74.26:9088/pinter/com/phone'
# json格式的参数通常以json来表示
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(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)
```

- delete接口

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2022-02-20 16:49
# @Copyright：北京码同学

import requests

url = 'http://82.156.74.26:9088/pinter/com/phone'
# json格式的参数通常以json来表示
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(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)
```

- cookie关联

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2022-02-20 17:02
# @Copyright：北京码同学

import requests

#这个session对象可以帮我们自动的关联cookie
#只要多个接口使用的是同一个session对象发起的接口调用，那么会自动关联cookie
session = requests.session() #这个session对象可以帮我们自动的关联cookie

# 先完成登录接口
url = 'http://82.156.74.26:9088/pinter/bank/api/login'

data = {
    "userName":"shamo",
    "password":"1253554"
}
resp = session.request(url=url,method='post',data=data)
status_code = resp.status_code #获取响应状态码
print(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)

# 调用查询余额的接口
url = 'http://82.156.74.26:9088/pinter/bank/api/query'
```

```
params = {
    "userName":"shamo"
}
resp = session.request(url=url,method='get',params=params)
status_code = resp.status_code #获取响应状态码
print(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)
```

- token关联

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2022-02-20 17:09
# @Copyright：北京码同学


import requests
#这个session对象可以帮我们自动的关联cookie
#只要多个接口使用的是同一个session对象发起的接口调用，那么会自动关联cookie
# 但是他不会帮我们自动关联token，token需要自行处理
session = requests.session() #这个session对象可以帮我们自动的关联cookie

# 先完成登录接口
url = 'http://82.156.74.26:9088/pinter/bank/api/login2'

data = {
    "userName":"shamo",
    "password":"1253554"
}
resp = session.request(url=url,method='post',data=data)
status_code = resp.status_code #获取响应状态码
print(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json) # {'code': '0', 'message': 'success', 'data':
'90e6488e44704f789936a36c8b2e5acf'}
# 从响应信息中提取data字段作为token
# resp.json()得到的是响应结果的字典形式，那么从字典中获取某个key对应的值
token = resp_json['data']

# 调用查询余额的接口
url = 'http://82.156.74.26:9088/pinter/bank/api/query2'

headers = {
    "testfan-token":token
}

params = {
    "userName":"shamo"
}
```

```
resp = session.request(url=url,method='get',headers=headers,params=params)
status_code = resp.status_code #获取响应状态码
print(status_code)
resp_info = resp.text #获取响应信息，返回结果是字符串格式
resp_json = resp.json()#获取响应信息，返回结果是个字典，注意这种方法只能针对响应信息是
json格式
print(resp_info)
print(resp_json)
```

- 买家token关联

```
# !/usr/bin python3
# encoding: utf-8 -*-
# @author: 沙陌 微信：Matongxue_2
# @Time: 2022-02-20 17:14
# @Copyright：北京码同学
import requests

session = requests.session()
token = ''
def buyer_login():
    url = 'http://www.mtxshop.com:7002/passport/login'

    headers = {
        'Authorization': ''
    }
    # 查询参数通常使用params来表示
    params = {
        'username': 'shamo',
        'password': 'e10adc3949ba59abbe56e057f20f883e',
        'captcha': '1512',
        'uuid': 'jsjdhdhdhdhdhdhh'
    }
    resp = session.request(url=url,method='post', headers=headers,
params=params)

    status_code = resp.status_code  # 获取响应状态码
    # print(status_code)
    # resp_info = resp.text  # 获取响应信息，返回结果是字符串格式
    resp_json = resp.json()  # 获取响应信息，返回结果是个字典，注意这种方法只能针对响
应信息是json格式
    # print(resp_info)
    print(resp_json)
    global token
    token = resp_json['access_token']
def add_cart():
    url = 'http://www.mtxshop.com:7002/trade/carts'
    global token
    headers = {
        'Authorization': token
    }
    # 查询参数通常使用params来表示
    params = {
        'sku_id': 5173,
        'num': 1
    }
```

```
    resp = session.request(url=url,method='post', headers=headers,
params=params)

    status_code = resp.status_code  # 获取响应状态码
    # print(status_code)
    # resp_info = resp.text  # 获取响应信息，返回结果是字符串格式
    resp_json = resp.json()  # 获取响应信息，返回结果是个字典，注意这种方法只能针对响
应信息是json格式
    # print(resp_info)
    print(resp_json)
if __name__ == '__main__':
    buyer_login()
    add_cart()
```