# [python接口自动化26-参数关联和JSESSIONID(上个接口返回数据作为下个接口请求参数)](https://www.cnblogs.com/yoyoketang/p/9671952.html)

# <https://www.cnblogs.com/yoyoketang/p/9671952.html>

# [python接口自动化24-有token的接口项目使用unittest框架设计](https://www.cnblogs.com/yoyoketang/p/9264773.html)

# <https://www.cnblogs.com/yoyoketang/p/9264773.html>

Python+Selenium面试题（附详细解答）

# <https://blog.csdn.net/maoxuexue/article/details/105898348?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-0&spm=1001.2101.3001.4242>

# 接口自动化测试面试题（1）



[candy\_tse\_1](https://blog.csdn.net/candy_tse_1) 2019-08-14 15:27:29  45086  收藏 771

分类专栏： [面试](https://blog.csdn.net/candy_tse_1/category_9208139.html) [接口测试](https://blog.csdn.net/candy_tse_1/category_9224217.html)

版权

根据网络资料，总结了以下一些常见的接口测试面试题：

1. 为什么要做接口测试？
2. 接口测试能发现哪些问题？
3. 接口测试怎么测？
4. 用什么工具测接口？
5. WebService接口是如何测试的？
6. 没有接口文档如何做接口测试？
7. 在接口测试过程中，上下游接口有数据依赖如何处理？
8. 依赖第三方数据的接口如何进行测试？
9. 当一个接口出现异常时，你是如何分析异常的？
10. 如何模拟弱网测试？
11. 如何分析一个bug是前端的还是后端的？

### 为什么要做接口测试

在讨论为什么要做接口测试之前，我们先稍微了解下接口是什么？

接口可以很不准确的理解成是与资源打交道，这个资源可能是本系统的，也可能是其他系统的。

举个例子，假如我们在开发1个bug管理系统，该系统需要拿到公司的所有开发和测试人员的信息，这样开发和测试人员不用注册都可以登录进去了，这应该很好理解。

那么这些人员的信息储存在哪里呢？一般存储在hr系统里。现在的需求更加明确了，我们要到hr系统中去拿到人员信息，获取hr系统中的人员资源。

怎么拿呢？很多种方式，可以直接把hr系统的数据库拷贝一份放到bug管理系统里，不过这样不好，因为数据的同步会有点麻烦；还可以直接连hr系统的数据库去查，这样也不太好，这样我们就需要了解hr系统的数据存储结构和逻辑，一旦hr系统的数据字段发生改变，bug管理系统也要去该，以便同步。

比较好的做法是，hr系统暴露一些接口，通过这些接口去获取人员信息资源，这样bug系统就不需要关心hr系统的数据存储实现了。

这些接口可能是这样的：

* 登录的接口，提供人员的用户名和密码，去hr系统中判断该人员是否存在，如果存在验证用户名和密码，如果验证通过就返回1个token，该token就是这个人员的通行证，通过token可以登录到bug管理系统中去；
* 获取人员信息的接口，返回该人员的职位：测试还是开发，以及用户名，昵称等信息；

综上：接口可以理解成是不同系统或模块之间资源交流方式。

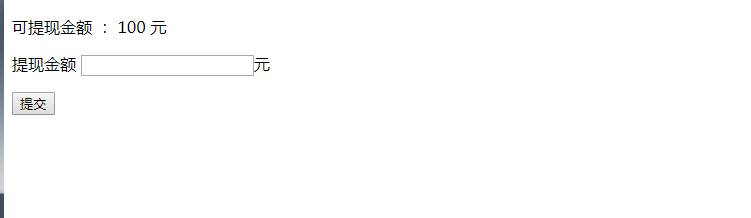
接口测试实际上是黑盒测试，基本的测试思路是根据输入和输出判断被测系统或对象的逻辑。获取人员的信息，我需要把人员的用户名传给hr系统接口，这样hr系统的接口会返回给我用户的一些更加具体的信息。这里的输入是用户名，输出是用户的详细信息。

既然是接口获取和操作资源的方式，而大部分系统和产品中，资源一般都是产品的核心，比如微信核心资源就是通讯录关系链和聊天记录等，因此资源是必测的。

另外接口中大部分的内容是数据，通过数据的对比我们能推测到系统和产品的逻辑，测接口就是测逻辑。

最后接口中的返回相对单纯，不像web页面，html代码中有太多ui的东西，ui最不稳定，变化太快，接口相对稳定一点点，但是里面的干扰信息更少，断言相对容易很多。

请看以下一个案例，如下图一个提现功能



比如这个输入框，平常拿到这个web页面，会对输入框做用例设计：

* 输入一个负数（如：-100），点提交
* 输入金额为0（如：0），点提交
* 输入金额为0-100的数（如：20），点提交
* 输入金额为100（如：100），点提交
* 输入金额大于100（如：108），点提交
* 输入1位小数（如：10.1），点提交
* 输入2位小数（如：10.12），点提交
* 输入3位小数（如：10.123），点提交

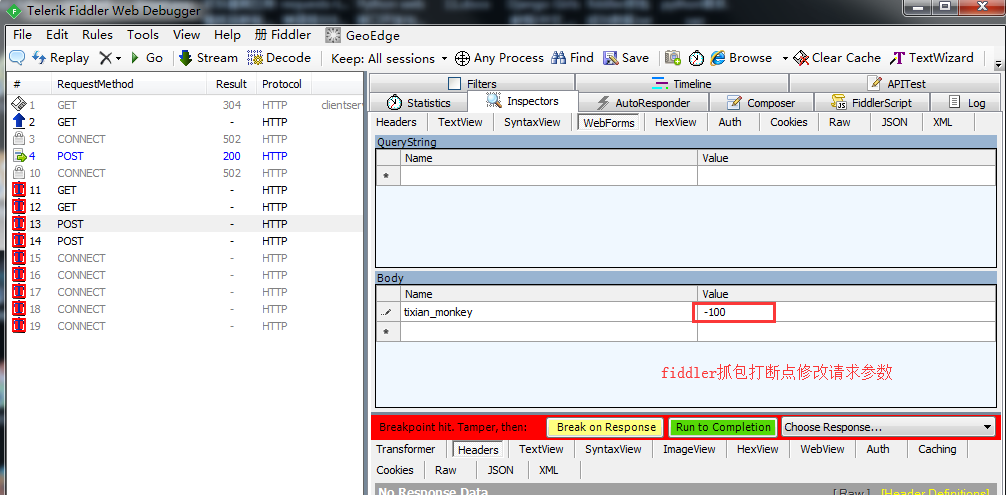
按照这个等价类，边界值用例测完，页面上不能输入负数和大于3位数小数点，然后就可以上线了。  
然而。。。突然有一天数据库里面插入了一个提现金额为负数（-100），于是整个部门炸锅了，首先找到测试（背锅）去复现问题，测试在页面上反复输入负数，无法提交，认为没问题啊！

首先前端开发对输入框是做了限制的，前端的web开发肯定没问题，这个锅前端开发MM不背。那么如果别人用户不通过你的web页面，直接发请求提交了呢？  
纳尼！！！不通过页面也能提交。。。这就是我们接下来要提到的接口测试了。

### 接口测试能发现哪些问题

这个问题其实回到起来很简单，只要做过接口测试的，总能发现几个BUG吧，把你平常发现的bug说2-3个就可以了。  
面试官出这个题，主要是想知道你是不是真的做过接口测试，毕竟现在很多小伙伴简历都是写的假的（你要不写估计面试机会都没有，没办法，为了生存，能理解）  
比如上面说的，提现输入框，在页面上输入负数，肯定是无法提交过去（前端页面会判断金额），如果我不走前端，直接用接口工具发请求，输入一个负数过去。  
（假设服务端没做提现金额数据判断）  
余额=当前余额（100）-提现金额（-100），那么提现-100，余额就变成200了，也就是越提现，余额越大了

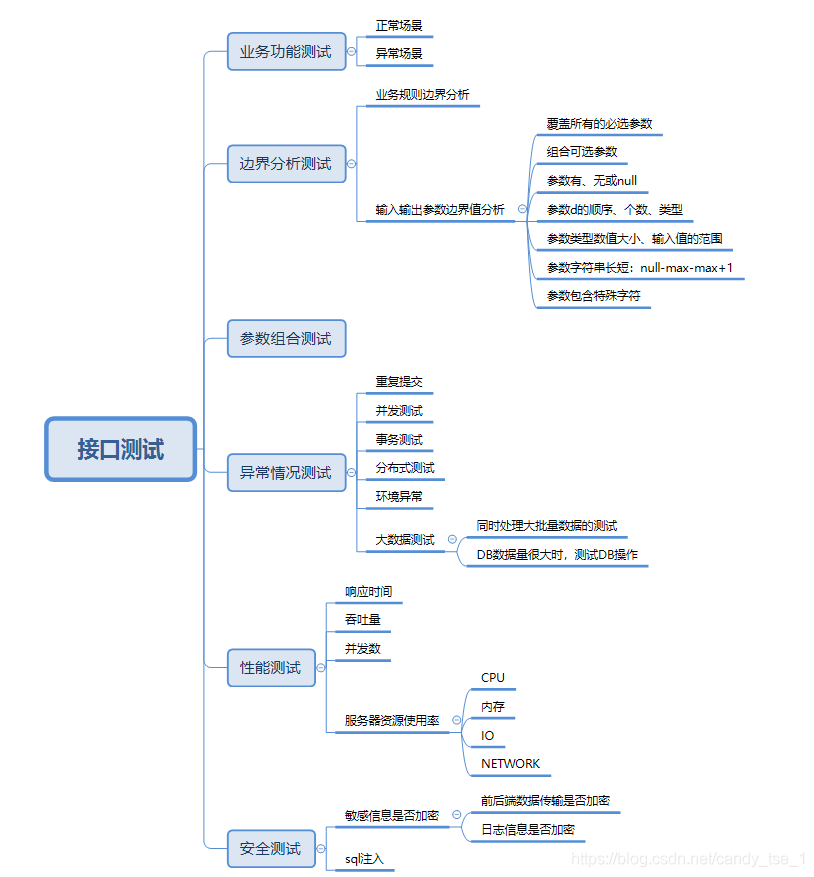
**可以用接口工具去直接请求接口，也可以fiddler抓包，抓到接口后修改金额为负数**



所以，接口测试的必要性就体现出来了：  
1.可以发现很多在页面上操作发现不了的bug  
2.检查系统的异常处理能力  
3.检查系统的安全性、稳定性  
4.前端随便变，接口测好了，后端不用变  
5.可以测试并发情况，一个账号，同时（大于2个请求）对最后一个商品下单，或不同账号，对最后一个商品下单  
6.可以修改请求参数，突破前端页面输入限制（如金额）

### 接口测试怎么测

* 通过性验证：首先肯定要保证这个接口功能是好使的，也就是正常的通过性测试，按照接口文档上的参数，正常传入，是否可以返回正确的结果。
* 参数组合：现在有一个操作商品的接口，有个字段type，传1的时候代表修改商品，商品id、商品名称、价格有一个是必传的，type传2的时候是删除商品，  
  商品id是必传的，这样的，就要测参数组合了，type传1的时候，只传商品名称能不能修改成功，id、名称、价格都传的时候能不能修改成功。
* 接口安全：  
  1、绕过验证，比如说购买了一个商品，它的价格是300元，那我在提交订单时候，我把这个商品的价格改成3元，后端有没有做验证，更狠点，我把钱改成-3，是不是我的余额还要增加？  
  2、绕过身份授权，比如说修改商品信息接口，那必须得是卖家才能修改，那我传一个普通用户，能不能修改成功，我传一个其他的卖家能不能修改成功  
  3、参数是否加密，比如说我登陆的接口，用户名和密码是不是加密，如果不加密的话，别人拦截到你的请求，就能获取到你的信息了，加密规则是否容易破解。  
  4、密码安全规则，密码的复杂程度校验
* 异常验证：  
  　　所谓异常验证，也就是我不按照你接口文档上的要求输入参数，来验证接口对异常情况的校验。比如说必填的参数不填，输入整数类型的，传入字符串类型，长度是10的，传11，总之就是你说怎么来，我就不怎么来，其实也就这三种，必传非必传、参数类型、入参长度。
* 性能测试  
  接口并发情况，如上面提到的：一个账号，同时（大于2个请求）对最后一个商品下单，或不同账号，对最后一个商品下单  
  接口响应时间，响应时间太长了，肯定需要优化，一般都是毫秒级别

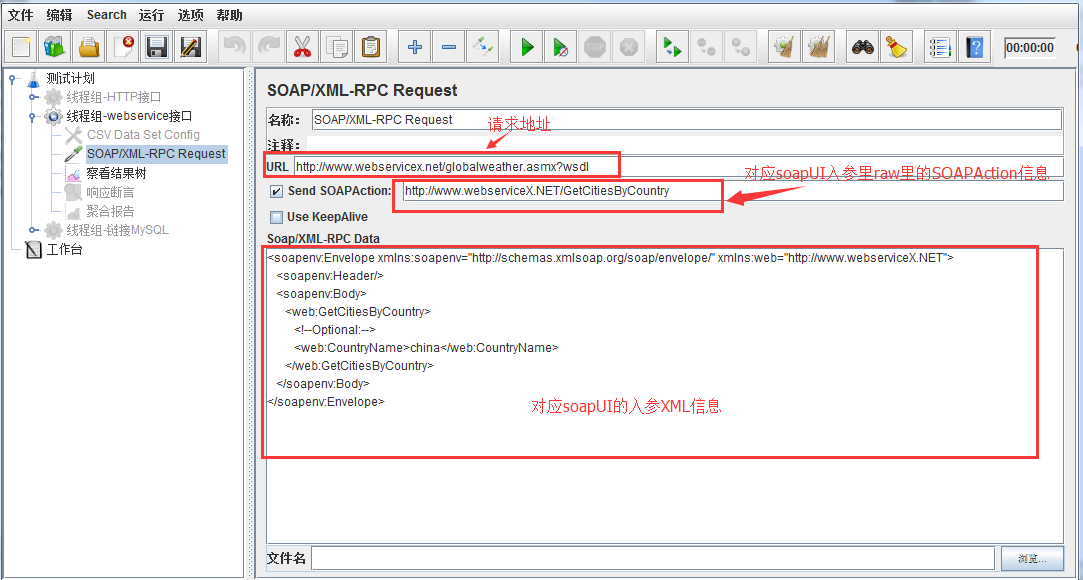


### 用什么工具测接口

* [postman](https://www.getpostman.com/)： 推荐。基本功能免费。最简单的基于http接口的调试和测试工具;
* [jmeter](http://jmeter.apache.org/)：后置处理器配合断言基本上可以满足接口测试需求，就是测试报告要做二次开发
* 自己撸代码：推荐。配合类似xunit测试框架，基本可以满足一切需求；[零基础实现python接口自动化视频教程，一起撸代码吧](https://ke.qq.com/course/232868#tuin=1eb87ef)
* [soapui](https://www.soapui.org/): 收费的；
* [insomnia](https://insomnia.rest/)：强力推荐。postman的弱化版，基本功能免费，重要的是工具代码开源，可以自己改；
* [paw](https://paw.cloud/): 强力推荐。mac上最强，淘宝买个授权好像就百把块钱；

### WebService接口是如何测试的

webService接口用SoapUI



### 没有接口文档如何做接口测试

没有接口文档，那还能咋办，瞎测呗！一个公司的开发流程里面，如果接口文档都没有，是无法展开接口测试的，你都不知道这个接口干什么的，也不知道具体每个字段代表什么意思，那还测啥呢？  
--当然，你肯定不能回答面试官不测(心理mmp，脸上笑嘻嘻)，接下来就是扯犊子时间  
1.没有接口文档，那就需要先跟开发沟通，然后整理接口文档（本来是开发写的，没办法，为了唬住面试官，先说自己整理了）  
2.没有接口文档，可以抓包看接口请求参数，然后不懂的跟开发沟通

本题主要靠情商，通俗来说就是忽悠能力，先唬住面试官了再说，进去了也是瞎测测，随时做好背锅的准备

### 在接口测试过程中，上下游接口有数据依赖如何处理

用一个全局变量来处理依赖的数据，比如登录后返回token,其它接口都需要这个token,那就用全局变量来传token参数

### 依赖第三方数据的接口如何进行测试

这个标准答案是：mock

接着面试官会问你，如果mock的，然后你就顺着坑继续挖，搭建mock服务，参考这篇[【https://www.cnblogs.com/yoyoketang/p/9348552.html】](https://www.cnblogs.com/yoyoketang/p/9348552.html)

### 当一个接口出现异常时，你是如何分析异常的

1.抓包，用fiddler工具抓包，或者浏览器上f12,app上的话，那就用fiddler设置代理，去看请求报文和返回报文了  
2.查看后端日志，xhell连上服务器，查看日志

### 如何模拟弱网测试

fiddler和charles都可以模拟弱网测试，平常说的模拟丢包，也是模拟弱网测试

### 如何分析一个bug是前端还是后端的

平常提bug的时候，前端开发和后端开发总是扯皮，不承认是对方的bug  
这种情况很容易判断，先抓包看请求报文，对着接口文档，看请求报文有没问题，有问题就是前端发的数据不对  
请求报文没问题，那就看返回报文，返回的数据不对，那就是后端开发的问题咯

**（1）我们测试的接口属于哪一类？**

      通常都是B/S架构，由客户端调用，或模拟客户端调用服务器提供的请求接口，由服务器完成处理并返回一个应答的过程。

      例如 webservice接口   http接口   jms接口   hessian接口

**（2）HTTP接口传递数据最常用的方式？**

      get和post最常用。

      get是从服务器上获取数据   适用场景：在做搜索接口  商品信息接口

      post是向服务器上传送数据   适用场景：在做数据添加修改删除时或图片上传等时能用到

**（3）HTTP接口测试的类型？**

      ① get请求无参数    ② get请求有参数     ③ post请求带参数

**（4）平常是怎么测接口的？**

      参数组合、接口安全(绕过前端验证、绕过身份授权、参数是否加密、密码安全规则)、异常验证(必传非必传、参数类型、入参长度)

      性能测试

      用postman工具测，其次用jmeter测试

**（5）webservice接口如何测试的？**

      用soapui工具测试

**（6）接口测试的步骤？**

      大体流程是：① 阅读API接口文档   ② 编写测试用例   ③ 根据测试用例进行API的手工测试(开发API工具、使用工具进行测试 )

**（7）没有文档如何做接口测试？**

      与开发沟通，然后整理出接口文档，也可以自己试着抓包看接口请求参数，然后不懂的跟开发沟通。

**（8）下个接口请求参数依赖上个接口的返回数据？**

      不同的接口封装成不同的函数或方法，需要的数据return出来，用一个中间变量a去接收。后面的接口传a就可以了。

**（9）在手工接口测试或者自动化接口测试过程中，上下游接口有数据依赖如何处理？**

      用一个全局变量来处理依赖的数据

**（10）接口测试中依赖登录状态的下的接口如何测试？**

      本质是：在每次发送请求时需要带上session或cookie才能发送成功，在构建post请求时添加必要的session或者cookie

**（11）依赖于登录的接口如何处理？**

      登录接口依赖token的，可以先登录后，token存到一个yaml或者json或者ini的配置文件里面

      后面所有的请求去拿这个数据就可以全局使用了

      如果是cookies的参数，可以用session自动关联

**（12）依赖第三方的接口如何处理？**

      这个需要自己去搭建一个mock服务、模拟接口返回数据

      mock是一个开源的框架，在github上可以下载到。

      mock可以用在单元测试过程中，还有一个用途，当前端开发在开发页面的时候，需要服务端提供API接口，此时服务端还没开发完成，或者说没搭建测试环境，这时候前端开发会自己mock一个api服务端，自己给自己提供调用接口的返回数据。

**（13）接口产生的垃圾数据如何清理？**

      造数据的数据清理，需用python连接数据库，做增删改查操作

      测试用例前置，setUp做数据准备

      后置操作，tearDown做数据清理

**（14）测试的数据你放在哪儿？**

      测试的数据分很多种，千万不能说写在代码里，因为数据是不能写死在代码里的。

      测试数据放到.py文件的开头，对于少量的，固定不变的数据是可以的。

      测试数据存放总结：

             ① 对于一些一次性消耗的数据，比如注册，每次注册不一样的数，可以用随机函数生成

             ② 对于一个接口有多组测试的参数，可以参数化，数据放yaml，text，json，execl等都可以

             ③ 对于反复使用的数据，如订单的各种状态需要造数据的情况，可以放到数据库，每次数据初始化，用完后再清理

             ④ 对于邮箱配置的一些参数，可以用ini配置文件

             ⑤ 对于全部是独立的接口项目，可以用数据驱动方式，用execl/csv管理测试的接口数据

             ⑥ 对于账号密码，这种管全局的参数，可以用命令行参数，单独抽出来，卸载i配置里（如ini）

             ⑦ 对于少量的静态数据，比如一个接口的测试数据，也就2-3组，可以写到python脚本的开头，十年八年都不会变更的。

**（15）json和字典的区别？**

     json是一种轻量级的数据交换格式。json本质上还是字符串，只是按key:value这种键值对的格式来的字符串。

     如：a = {'a':1, 'b': 2}  这是字典

           a = '{'a': 1, 'b': 2}'  这是json

**（16）不可逆操作，如何处理，比如删除一个订单这种接口如何测试？**

     主要靠手工测试造数据，直接在数据库改字段状态。自动化也一样，造数据可以用python连接数据库，做增删改查的操作

     测试用例前置，setUp做数据准备

     后置操作，tearDown做数据清理

**（17）一个订单的几种状态如何全部测到。如：未处理、处理中、处理失败、处理成功？**

     也是一样，造数据，通过python连接数据库，修改数据的状态

**（18）python如何连接数据库操作？**

     PyMYSQL是在python3.x版本中用于连接mysql服务器的一个库

     mysqldb是在python2中使用的

      - 首先导入模块(提前pip安装)  import pymysql  或者 import mysqldb

      - 打开数据库连接   db = pymysql.connect("localhost", "username", "psw", "db\_name")

      - 创建一个游标对象  cursor = db.cursor()

      - sql查询语句   sql = "select \* from emp"

      - 执行sql语句  cursor.execute(sql)

      - 获取所有记录列表   cursor.fetchall()  然后for循环遍历

      - 关闭数据库连接  db.close()

**（19）做接口测试过程中发现过哪些bug？**

比如：测试体现金额的提交，有一天数据库里插入了一个提现金额为负数的，开发要你复现，你在页面上怎样测试都没错误，那是前端做了限制。可以绕过前端，直接用接口工具去直接请求接口，或者用fiddler去抓包，抓到接口劶修改金额为负数。

**（20）当一个接口出现异常，你是如何分析异常的？**

      用fiddler工具抓包或者浏览器上按F12调试。app上就用fiddler设置代理，去看请求报文和返回报文。

      查看后端日志，xshell连上服务器，查看日志。

**（21）如何模拟弱网设置？**

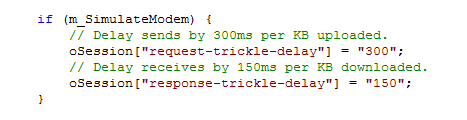
      fiddler和charles都可以模拟，平常说的模拟丢包，也就是模拟弱网测试。

      通过charles设置延迟，进行模拟不同的网络：配置好charles后，正常联网，选择throttle settings设置弱网环境

          thtottle preset选择弱网环境目标：2G或3G；也可以在Bandwidth中选择上传下载数值(kbps)

      通过fiddler模拟弱网：打开fiddler ->rules ->customer rules ->会出现一个fiddler script编辑器窗口，

           利用ctrl+F组合键调出搜索对话框，键入m\_Simulate进行搜索，找到如下代码框：



           upload代表上传速度，download代表下载速度。自定义修改之后，保存后退出。

           继续在fiddler界面上找到rules-> perfomance-> 点击Simulate Modem Speeds，完成弱网模拟功能的打开。

**（22）怎么分析bug是前端的还是后端的？**

      先抓包看请求报文，对着接口文档，看请求报文有没问题，有问题就是前端发的数据不对。

      请求报文没问题，那就看返回报文，返回的数据不对，那就是后端开发的问题。

**（23）http接口和web service接口测试区别是什么？**

      传统web service：  有一套完整的协议标准，其中有soap协议，用来进行消息的传递。

            soap请求是HTTP POST的一个专用版本，遵循一种特殊的xml消息格式  Content-type设置为：text/xml任何数据都可以xml化。

            web service的好处是：不用担心大小写问题

                                                 不用担心中文urlencode问题

                                                 代码中不用多次申明认证(账号、密码)参数

                                                  传递参数可以为数组、对象等

             由于web service要进行xml解析，速度可能会有所降低。

      现在HTTP完全可以替代web service。

1.什么是接口测试？

* **接口测试**：是测试系统组件间接口的一种测试方法
* **接口测试的重点**：检查数据的交换，数据传递的正确性，以及接口间的逻辑依赖关系
* **接口测试的意义**：在软件开发的同时实现并行测试，减少页面层测试的深度，缩短整个项目的测试周期

2.接口自动化测试的流程？

基本的接口功能自动化测试流程为：需求分析-->用例设计-->脚本开发-->测试执行-->结果分析

3.GET请求和POST请求区别是什么？

* **GET请求**：用于信息获取，相对而言是安全和幂等的；在做数据查询时，建议用GET方式，如：商品信息接口、搜索接口、博客访客接口...
* **POST请求**：表示可能会修改服务器上资源的请求；在做数据添加、修改时，建议用POST方式。如：上传图片接口、登录注册接口...

**误区：**  
*“GET是从服务器上获取数据，POST是向服务器传送数据”*（该说法有误）  
**解析：**  
GET/POST都可以提交数据，GET请求也可以向服务器传递数据，POST请求也需要服务器返回数据

4.接口测试的常用工具有哪些？

* Postman
* JMeter
* SoapUI

5.HTTP接口的请求参数类型有哪些？

* 查询字符串参数（Query String Parameters参数）一般用于GET请求，会以url string的形式进行传递
* 请求体参数（Request Body）一般用于POST请求，可以使用Content-Type来指定不同参数类型

6.如何从上一个接口获取相关的响应数据传递到下一个接口？

先从上一个接口中的响应数据获取对应的返回值，然后使用正则表达式or使用JSON解析来提取需要获取的值，然后存储在一个变量中，最后在下一个接口中直接引用该变量即可

7.接口测试用例的编写要点有哪些？

1）必填字段：请求参数必填项、可选项  
2）合法性：输入输出合法、非法参数  
3）边界：请求参数边界值等  
4）容错能力：大容量数据、频繁请求、重复请求（如：订单）、异常网络等的处理  
5）响应数据校验：断言、数据提取传递到下一级接口...  
6）逻辑校验：如两个请求的接口有严格的先后顺序，需要测试调转顺序的情况  
7）性能：对接口模拟并发测试，逐步加压，分析瓶颈点  
8）安全性：构造恶意的字符请求，如：SQL注入、XSS、敏感信息、业务逻辑（如：跳过某些关键步骤；未经验证操纵敏感数据）

8.接口测试的步骤有哪些？

1）发送接口请求  
2）测试接口获取返回值  
3）断言：判断实际结果是否符合预期

9.接口测试中依赖登录状态的接口如何测试？

依赖登最状态的接口，本质上是在每次发送请求时需要带上存储有账户有效信息的Session或Cookie才能发送成功，在构建POST请求时添加必要的Session或Cookie

10.依赖于第三方数据的接口如何进行测试？

可以利用一些MOCK工具（如：JSON Server、Easy Mock）来模拟第三方的数据返回，最大限度的降低对第三方数据接口的依赖

1. Dd

## 什么是API？

API是(Application Programming Interface)首字母缩略词，即应用程序编程接口。 API是一组用于构建软件应用程序的规程，协议和工具。API充当软件应用程序之间的接口，并允许两个软件应用程序相互通信。 API是一组软件功能，可以由其他软件执行。

## 什么是API测试？

API测试是一种软件测试，涉及直接测试API，也是集成测试的一部分，用于检查API是否满足应用程序的功能，可靠性，性能和安全性方面的期望。在API测试中，我们主要关注软件架构的业务逻辑层。可以在包含多个API的任何软件系统上执行API测试。

## 常见的API测试类型有哪些？

API测试通常涉及以下实践：

* 单元测试
* 功能测试
* 负载测试
* 运行时/错误检测
* 安全测试
* UI测试
* 互操作性和WS一致性测试
* 渗透测试
* 模糊测试

## 列举API测试中使用的一些常用协议？3个即可。

* thrift
* HTTP
* REST
* SOAP
* JMS
* UDDI
* dubbo

## API和Web服务之间的区别？

Web服务：

* 所有Web服务都是API
* 所有Web服务都需要通过Web(HTTP)公开
* Web服务只有三种使用方式：SOAP，REST和XML-RPC进行通信

接口：

* API有很多并不基于HTTP
* API使用多种方式进行通信，例如C / C ++中的DLL文件，java中的Jar文件/ RMI，Linux内核API中的中断等。

## 什么是Soap？

SOAP代表简单对象访问协议(Simple Object Access Protocol)。它是一种基于XML的消息传递协议。虽说名字带了简单，但是协议比较罗嗦，已经远没有后来居上的JSON使用广泛。

## 什么是Rest API？

REST即Representational State Transfer。它是一组帮助开发人员执行请求和接收响应的函数。通过REST API中的HTTP协议进行交互。

## SOAP和REST的区别？

SOAP：

* 通过共享XML文档进行通信
* 仅支持XML格式
* 不支持缓存
* SOAP比REST慢
* SOAP就像自定义桌面应用程序，紧密连接到服务器
* SOAP基于HTTP进行封装

REST： - 基于网络的软件架构的服务架构和设计 - 支持不同的数据格式 - 支持缓存 - 比SOAP更快 - REST客户端就像浏览器并使用应用程序必须适合的标准方法 - REST使用HTTP标头来保存元信息

## API常见测试有哪些？

我们在API上执行的一些常见测试如下。

* 验证不同输入条件的返回。
* 验证不同数据结构
* 验证API是否触发其他事件或请求其他API
* 在没有返回值时验证API的行为

## API测试有哪些优势？

* 更快及更高的测试覆盖率。
* API测试有助于我们降低测试成本。通过API测试，我们可以在GUI测试之前找到小错误。在GUI测试期间，这些小错误将变得更大。因此，在API测试中发现这些错误将对公司具有成本效益。
* API测试与语言无关。
* API测试在测试核心功能方面非常有用。我们可以在没有用户界面的情况下测试API。在GUI测试中，我们需要等到应用程序可用于测试核心功能。
* API测试有助于我们降低风险。

## API测试中究竟需要验证哪些内容？

* 数据准确性
* HTTP或其他协议状态代码
* 响应时间
* API返回任何错误时的错误代码
* 授权检查
* 非功能测试，如性能测试，安全测试

## 列举一些用于API测试的工具？

用于API测试的一些工具如下：

* [Curl](https://link.zhihu.com/?target=https%3A//github.com/curl/curl)
* [httpie](https://link.zhihu.com/?target=https%3A//github.com/jakubroztocil/httpie)
* [Requests: HTTP for Humans](https://link.zhihu.com/?target=http%3A//docs.python-requests.org/en/master/)
* [https://pypi.org/project/thrift/](https://link.zhihu.com/?target=https%3A//pypi.org/project/thrift/)
* [Postman](https://link.zhihu.com/?target=https%3A//www.getpostman.com/)
* [Katalon Studio](https://link.zhihu.com/?target=https%3A//www.katalon.com/)
* [SoapUI](https://link.zhihu.com/?target=https%3A//www.soapui.org/)
* [Assertible](https://link.zhihu.com/?target=https%3A//assertible.com/)
* [Tricentis Tosca](https://link.zhihu.com/?target=https%3A//www.tricentis.com/software-testing-tools/)
* [Apigee](https://link.zhihu.com/?target=https%3A//apigee.com/)
* [JMeter](https://link.zhihu.com/?target=https%3A//jmeter.apache.org/)
* [Rest-Assured](https://link.zhihu.com/?target=http%3A//rest-assured.io/)
* [Karate DSL](https://link.zhihu.com/?target=https%3A//github.com/intuit/karate)
* [API Fortress](https://link.zhihu.com/?target=http%3A//apifortress.com/)
* [Parasoft](https://link.zhihu.com/?target=https%3A//www.parasoft.com/)
* [HP QTP(UFT)](https://link.zhihu.com/?target=https%3A//software.microfocus.com/)
* [vREST](https://link.zhihu.com/?target=https%3A//vrest.io/)
* [Airborne](https://link.zhihu.com/?target=https%3A//github.com/brooklynDev/airborne)
* [API Science](https://link.zhihu.com/?target=https%3A//www.apiscience.com/)
* [APIary Inspector](https://link.zhihu.com/?target=https%3A//help.apiary.io/tools/api-inspector/)
* [Citrus Framework](https://link.zhihu.com/?target=https%3A//citrusframework.org/)
* [Hippie-Swagger](https://link.zhihu.com/?target=https%3A//github.com/CacheControl/hippie-swagger)
* [HttpMaster Express](https://link.zhihu.com/?target=https%3A//www.httpmaster.net/)
* [Mockbin](https://link.zhihu.com/?target=http%3A//mockbin.org/)
* [Ping API](https://link.zhihu.com/?target=https%3A//ping-api.com/)
* [Pyresttest](https://link.zhihu.com/?target=https%3A//github.com/svanoort/pyresttest)
* [Rest Console](https://link.zhihu.com/?target=https%3A//github.com/ahmadnassri/restconsole)
* [RoboHydra Server](https://link.zhihu.com/?target=http%3A//robohydra.org/)
* [SOAP Sonar](https://link.zhihu.com/?target=http%3A//www.crosschecknet.com/products/soapsonar.php)
* [Unirest](https://link.zhihu.com/?target=http%3A//unirest.io/)
* [WebInject](https://link.zhihu.com/?target=http%3A//www.webinject.org/)

## 列出一些最常用的API文档模板？

一些API文档模板如下。

* Swagger
* FlatDoc
* RestDoc
* API blueprint
* Slate
* Miredot
* Web服务API规范。

## 列举一些非常受欢迎的API。

* Google Maps API
* YouTube的
* 推特Twitter
* 亚马逊广告API
* 微博认证API
* 钉钉开放接口

## API测试和单元测试之间的区别？

单元测试： - 多由开发团队进行 - 白盒测试 - 构建中的过程之前 - 涉及源代码 - 测试范围有限，只考虑基本功能

API测试

* 多由QA团队进行
* 多为黑盒测试
* 在构建部署后进行
* 大多不涉及源代码API测试
* 测试范围很广

## API测试面临的主要挑战是什么？

* 适当的参数及其组合
* 正确分类参数
* 顺序
* 验证输出
* 由于缺少GUI，提供输入值较困难

## 执行API测试时我们面临的BUG类型是什么？

* 压力，性能和安全问题
* 功能重复或缺失
* 可靠性问题
* 消息不当
* 不兼容的错误处理机制
* 多线程问题
* 不合适的错误

## UI测试与API测试有何不同？

UI（用户界面）测试是测试应用程序的图形界面部分。它的主要重点是测试应用程序的外观和感觉。API测试支持两个不同软件系统之间的通信。它的主要重点是应用程序的业务层。

## 列举一些最常用的HTTP方法？

* GET：从服务器检索数据
* POST：将数据添加到服务器中的现有文件或资源
* PUT：它允许您替换服务器中的现有文件或资源
* DELETE：它允许您从服务器中删除数据
* PATCH：用于对资源进行部分修改 选项：用于描述目标资源的通信选项
* HEAD：它要求响应与GET请求相同，但没有响应正文

## 可以使用GET请求而不是PUT来创建资源吗？

不，GET请求仅允许只读权限。它使您可以从服务器检索数据，但不能创建资源。应使用PUT或POST方法来创建资源。

## PUT和POST方法有什么区别？

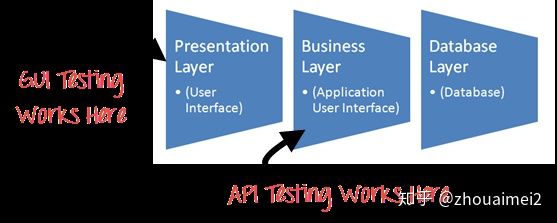
POST用于在服务器上创建新对象，PUT请求用于在替换对象。

当客户端将页面发送到服务器，然后服务器让客户端知道它放在何处时，应该使用POST。当客户端指定页面的位置时，应使用PUT

## 

**什么是API**  
在进行Api测试之前，让我们先了解一下什么是API？  
API是应用程序编程接口(Application Programming Interface)的首字母缩写。  
它支持两个独立软件系统之间的通信和数据交换。实现API的软件系统包含可以由其他软件系统执行的功能/子程序。  
什么是API测试？  
API测试与GUI测试完全不同，主要集中在软件架构的业务逻辑层。这种测试不太关注应用程序的外观和感觉。  
在API测试中，您不必使用标准用户输入（键盘）和输出，而是使用软件将调用发送到API，获取输出并记下系统的响应。  
在API测试中测试需要应用程序与API进行交互。为了测试API，你需要

* 使用测试工具来驱动API
* 编写你自己的代码来测试API



设置API测试环境

* API测试与其他测试类型不同，因为GUI不可用，但您需要设置初始环境，以调用具有所需参数集的API，然后最终检查测试结果。
* 因此，为API测试设置测试环境似乎有点复杂。
* 数据库和服务器应按照应用程序要求进行配置。
* 安装完成后，应调用API函数来检查该API是否正常工作。

API的输出类型  
API的输出可能是

* 任何类型的数据

例如：有一个API函数应该为两个整数求和：  
Long add（int a，int b）  
数字必须作为输入参数给出。输出应该是两个整数的总和。这个输出需要用预期的结果进行验证。  
调用比如  
add (1234, 5656)  
如果超过整数限制，则必须处理例外情况。

* 状态（例如Pass或Fail）

考虑下面的API函数:  
Lock() Unlock() Delete()  
它们返回任何值，如True（成功的情况下）或False（如果有错误）作为输出。  
更精确的测试用例可以调用任何脚本中的函数，并在数据库或应用程序GUI中中检查更改。

* 调用另一个API函数。

例如 - 第一个API函数可用于删除表中的指定记录，而此函数又调用另一个函数来刷新数据库。  
API测试的测试用例：

* 基于输入条件的返回值：测试相对容易，因为可以定义输入并且可以验证结果
* 不返回任何内容：当没有返回值时，检查系统上API的行为
* 触发其他API/事件/中断：如果API的输出触发某个事件或中断，则应跟踪这些事件和中断侦听器
* 更新数据结构：更新数据结构会对系统产生一些结果或影响，并且应该进行认证
* 修改某些资源：如果API调用修改了一些资源，则应通过访问各自的资源来验证它

API测试方法：

* 了解API程序的功能并明确定义程序的范围
* 用等效类，边界值分析和错误猜测等测试技术，并为API编写测试用例
* API的输入参数需要进行适当的规划和定义
* 执行测试用例并比较预期结果和实际结果。

API测试和单元测试的区别  
单元测试 API测试 多为开发人员执行 多为测试人员执行 独的功能 端到端功能 以访问源代码 经常无法访问源代码 涉及到UI测试 只有API 通常只有基本功能 通常涉及所有功能 范围有限 范围更广 通常在check-in前运行 builid完成后运行 在API测试中测试什么  
除了通常的SDLC（Software Development Life Cycle 软件生命周期）过程外，API测试应覆盖至少以下测试方法

* 发现测试：测试组应手动执行API中记录的一组调用，例如验证是否可以产看，创建和删除由API公开的特定资源
* 可用性测试：验证API是否可用和用户友好。 API是否与另一个平台完美集成
* 安全测试：此测试包括需要哪种类型的身份验证以及敏感数据是通过HTTP进行加密还是两者兼而有之
* 自动化测试：API测试应最终创建一组脚本或可用于定期执行API的工具
* 文档：测试团队必须确保文档足够，并提供足够的信息与API进行交互。文档应该是最终可交付成果的一部分

API测试的最佳实践：

* 测试用例应按测试类别进行分组
* 在每个测试的最上面，你应该包括被调用的API的声明。
* 参数选择应该在测试用例本身中明确提及
* 优先考虑API函数调用，以便测试人员能够轻松测试
* 每个测试用例应该尽可能独立
* 避免在开发过程中使用“测试链”
* 处理一次调用函数时必须特别小心，如 - Delete，CloseWindow等等。
* 调用队列
* 为确保完整的测试覆盖率，请为API的所有可能输入组合创建测试用例。

测试检测到的错误类型

* 无法正常处理错误条件。比如分母为0、文件不存在、忘记导入库等。
* 未使用的标志
* 缺少或重复的功能
* 可靠性问题。连接并获取API响应时遇到困难。
* 安全问题
* 多线程问题
* 性能问题。 API响应时间非常高。
* 不正确的错误/警告给调用者
* 错误的有效参数值处理
* 响应数据结构不正确（JSON或XML）

API测试工具  
由于API和单元测试都针对源代码，因此可以使用类似的工具进行测试。

* SOAPUI - 这是一个易于使用的工具，具有广泛的REST和SOAP API测试功能 - 无需脚本经验。
* [Runscope](https://link.zhihu.com/?target=https%3A//www.runscope.com/)
* [Postman](https://link.zhihu.com/?target=https%3A//www.getpostman.com/)
* [Curl](https://link.zhihu.com/?target=https%3A//github.com/curl/curl)
* [httpie](https://link.zhihu.com/?target=https%3A//github.com/jakubroztocil/httpie)
* [cfix](https://link.zhihu.com/?target=http%3A//www.cfix-testing.org/unit-testing-framework/)
* Check
* CTESK
* dotTEST
* Eclipse SDK工具 - 自动API测试
* [更多测试工具](https://link.zhihu.com/?target=https%3A//github.com/china-testing/python-api-tesing%23%25E6%25B5%258B%25E8%25AF%2595%25E6%25A1%2586%25E6%259E%25B6)
* [Top 20 API Testing Tools In 2018: REST & SOAP](https://link.zhihu.com/?target=https%3A//www.guru99.com/top-6-api-testing-tool.html)

技术支持qq群：测试开发自动化测试 144081101 Python自动化测试初学者 567351477  
API测试的挑战

* API测试中的主要挑战是参数组合，参数选择和调用排序
* 没有GUI可用于测试难以给出输入值的应用程序
* 验证和验证不同系统中的输出对于测试人员来说很难
* 测试人员需要了解参数选择和分类
* 异常处理功能需要测试
* 编码知识对于测试人员是必要的

结论：  
API由代表业务逻辑层的一组类/功能/程序组成。如果没有正确测试API，则不仅会导致API应用程序出现问题，而且还会导致在调用应用程序中出现问题。

**jmeter做接口测试（值得一看）**



[蒲公英上的尘埃](https://blog.csdn.net/loner_fang) 2018-08-06 17:42:40  14899  收藏 56

分类专栏： [接口测试](https://blog.csdn.net/loner_fang/category_7823922.html)

**JMeter对各种类型接口的测试**

默认做接口测试前，已经给出明确的接口文档（如，<http://test.nnzhp.cn/wiki/index.php?doc-view-59>）；本地配好了JMeter 3.x的运行环境；

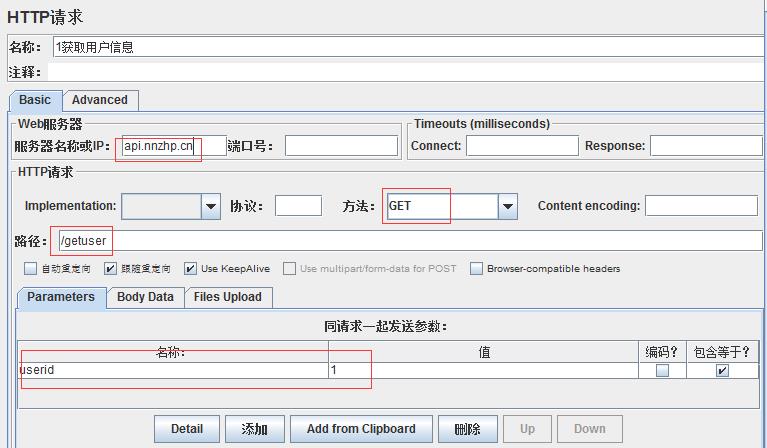
打开JMeter，添加一个线程组和该线程组的查看结果树。以下的几种接口请求我们都在这个线程组中添加和运行。

如果你的JMeter返回数据是乱码，解决办法是，在JMeter安装路径的bin目录下，打开文件jmeter.properties，把Sampleresult.default.encoding的值改为 utf-8 即可。

1. 普通的以key-value传参的get请求

e.g. 获取用户信息

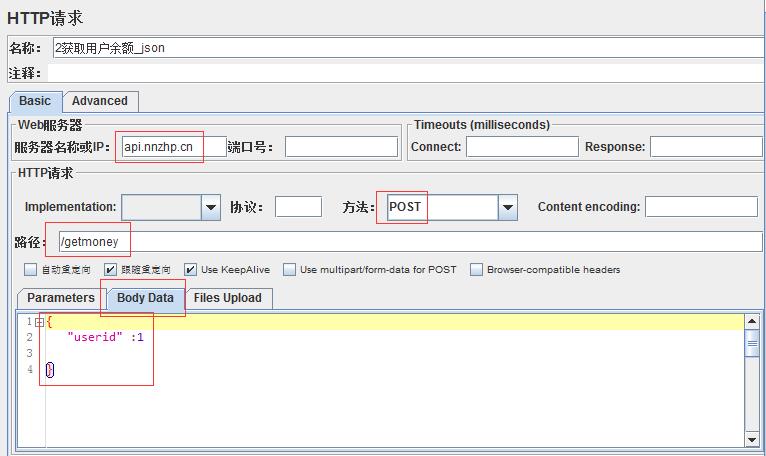
添加http请求；填写服务器域名或IP；方法选GET；填写路径；添加参数；运行并查看结果。



 2. 以Json串传参的post请求

e.g. 获取用户余额

添加http请求；填写服务器域名或IP；方法选POST；填写路径；在参数区域选Body Data标签；写入Json串参数，注意Json格式；运行并查看结果。



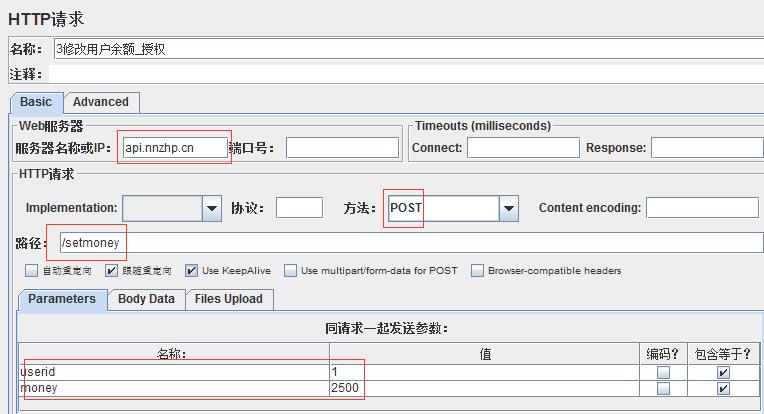
**\*注意：**

Json传参时，往往参数相对复杂，规模较大，很多时候是Json串再嵌套Json串，在写入body的时候如果哪里格式不对会影响传参，而JMeter本身是不能对Json进行格式校验的（Postman可以进行简单的Json校验），我们可以借助这个网站来校验你的Json串的格式，[www.bejson.com](http://www.bejson.com/) 再把确保格式正确的Json串填入body。

3.  请求时需要http权限验证

e.g. 修改用户余额

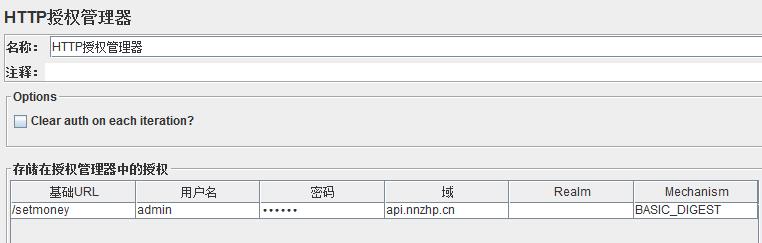
添加http请求；填写服务器域名或IP；方法选POST；填写路径；填写参数；



添加授权管理器，右击线程组->添加->配置元件->HTTP授权管理器；



打开HTTP授权管理器，在基础URL添加路径；填写用户名、密码（接口文档会给出）；添加域即请求的服务器域名或IP；其他默认；

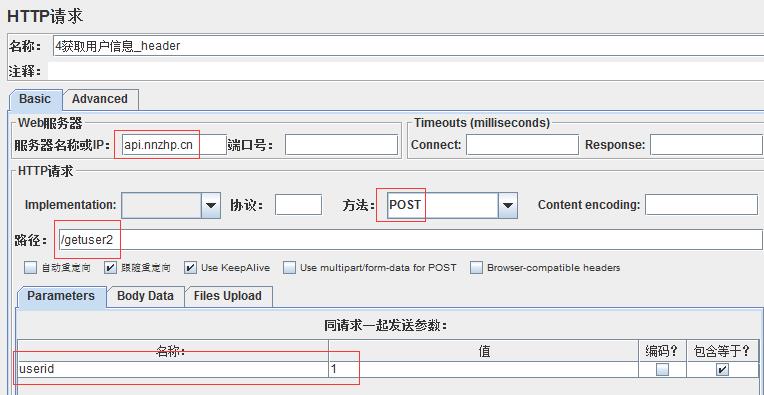


运行请求并查看结果。（该接口暂时有问题请求返回500，只要掌握需要权限验证的接口如何做就好）

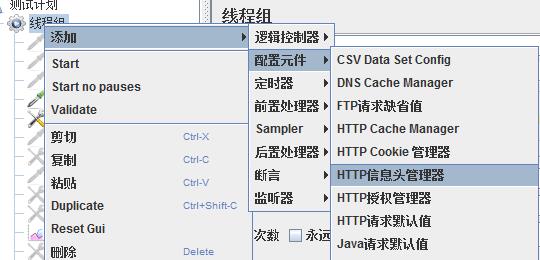
4.  请求时需要添加请求标头

e.g. 获取用户信息2

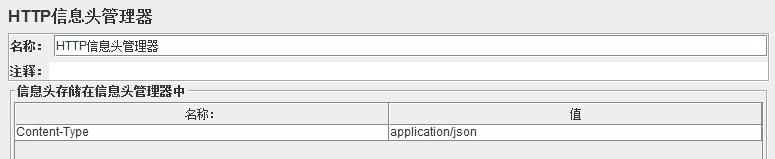
添加http请求；填写服务器域名或IP；方法选POST；填写路径；填写参数；



添加HTTP信息头管理器，右击线程组->添加->配置元件->HTTP信息头管理器；



打开HTTP信息头管理器，参照接口文档填写好header信息；

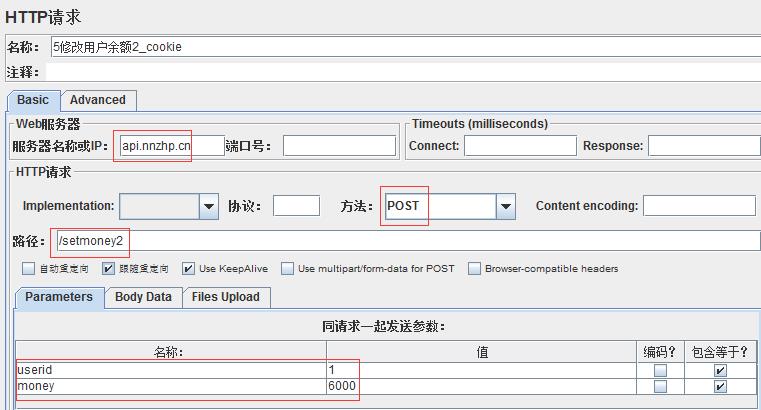


运行请求并查看结果。

5.  请求时需要添加Cookie

e.g. 修改用户余额2

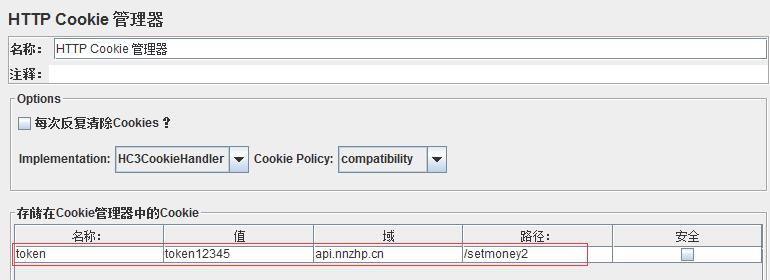
添加http请求；填写服务器域名或IP；方法选POST；填写路径；填写参数；



添加HTTP Cookie管理器，右击线程组->添加->配置元件->HTTP Cookie管理器；



打开Cookie管理器，参照接口文档填写好Cookie信息；

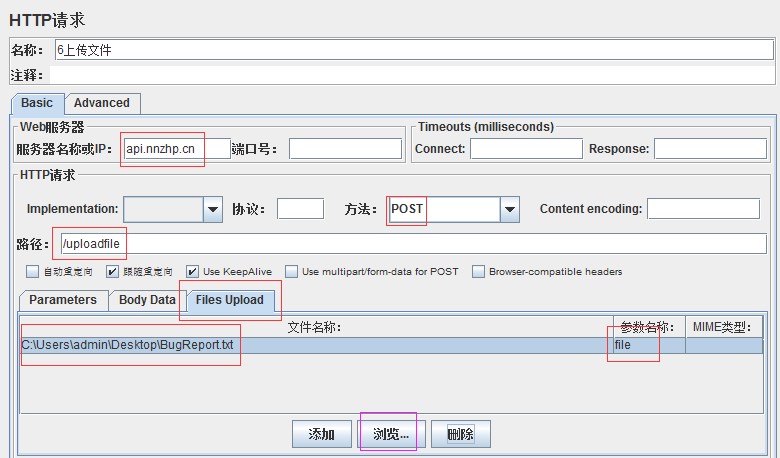


 发送请求并查看结果。

6.  请求通过某个接口上传文件

    e.g.上传文件

    添加http请求；填写服务器域名或IP；方法选POST；填写路径；在参数区域选Files Upload标签，添加文件全名（包含绝对路径，也可以点击浏览按钮添加），写入参数名；发送请求并查看结果。



**JMeter的参数化**

1. 指定参数值

这种方法常用来解决线程组中多个请求共用相同的数据，比如我们以上所有请求都有相同的IP地址，当地址改变时可以一改全改。

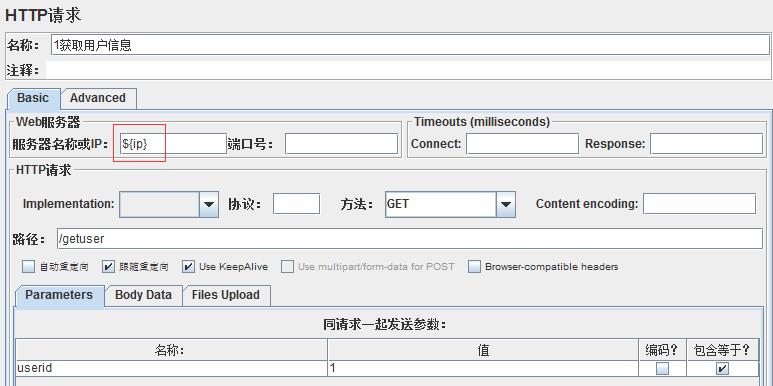
添加用户定义的变量，右击线程组->添加->配置元件->用户定义的变量；



打开用户定义的变量，添加变量，写好名称和值；



脚本中用到该值的所有地方都可以引用这个变量名了，引用变量的格式为 ${变量名}；



2. 借助函数生成参数

测试过程中，有时候数据不适合被指定，可选择借助函数来生成，以下是最常用的几种。

* 用函数生成随机数：（\_\_Random）

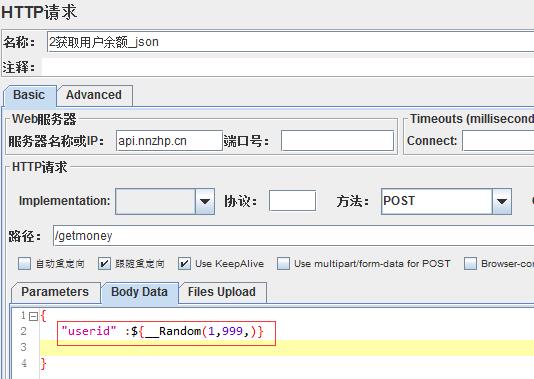
在Jmeter中点击 选项->函数助手对话框；



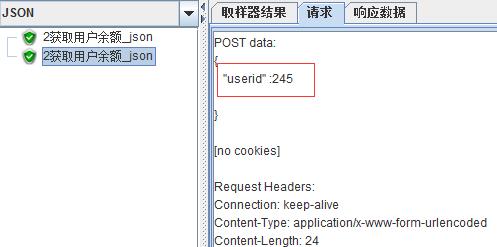
在弹出的对话框中选\_\_Random，填写随机数的范围，点击生成按钮，就看到你需要的字符串了；



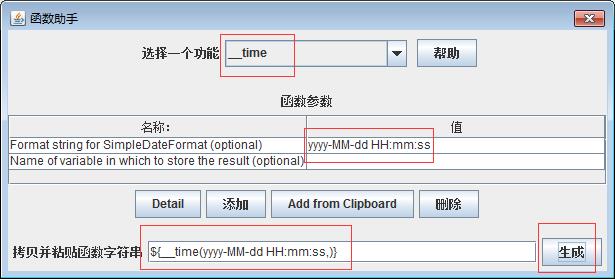
尝试在请求中使用生成的随机数，把刚拷贝的字符串粘贴到请求的参数里；



发送请求，查看请求中参数值的情况，参数生效了。



* 用函数生成当前时间：（\_\_time）

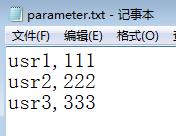


* 用函数生成通用的唯一识别码：（\_\_UUID）

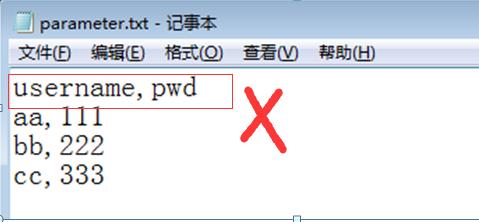


3. 文件传参

用于读取在文件中维护的参数，如参数文件内容如下；



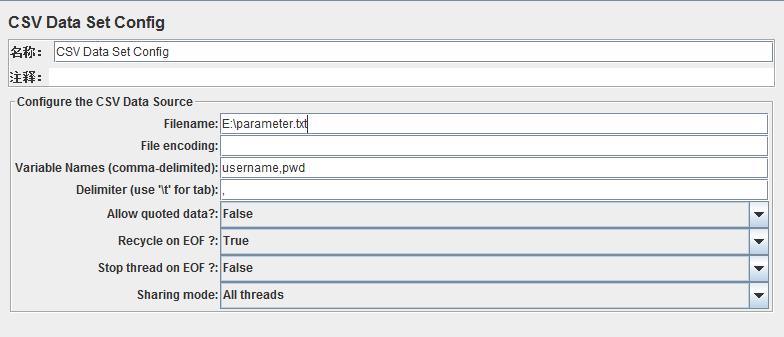
注意，Jmeter读取文件总的参数是从第一行就开始了的，千万不要犯下面的错误。



在Jmeter中右击线程组->添加->配置元件->CSV Data Set Config；



打开设置窗口，写入参数文件的全名（含绝对路径），变量名，Delimiter就用默认的逗号，表示在准备参数文件时参数表的列之间用逗号隔开；



在请求中引用参数时应该是酱紫的：${username}  ${pwd}，用它们来顶替对应的参数值。

4. 关联传参

在线程组内，当你想用一个请求的响应结果作为另一个请求的入参时就需要用到关联。

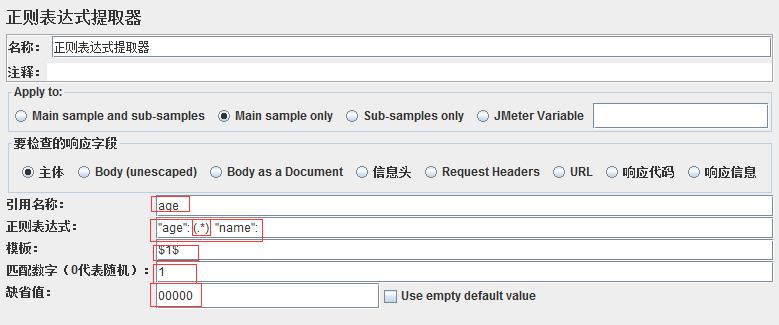
* 用正则表达式关联

思路是先从某个请求的响应数据中提取你需要的值，在把这个值在另一个请求中入参，操作如下：

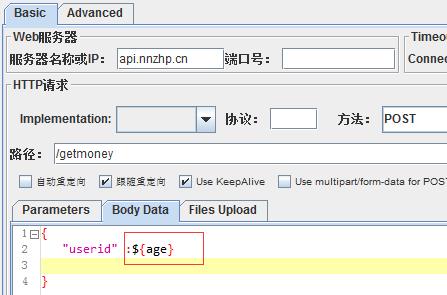
在需要被提取响应数据的请求下添加正则表达式提取器，右击添加->后置处理器->正则表达式提取器；



在正则表达式提取器中填写 引用名称，不说也知道用途；正则表达式，最简单的办法是在响应数据中把要提取的目标值左右若干字符包含目标值的一行拷贝出来，粘贴到正则表达式框内，在把目标值用一对小括号替换，括号里用添上合适的匹配符本例如(.\*)； 注意在响应数据中拷贝时尽量避开需要转义的字符如果不能避开，请转义；模板通常固定填$1$；匹配数字1代表区匹配的第一个值；缺省值是在匹配失败时取得的值，通常用来检查匹配是否成功；



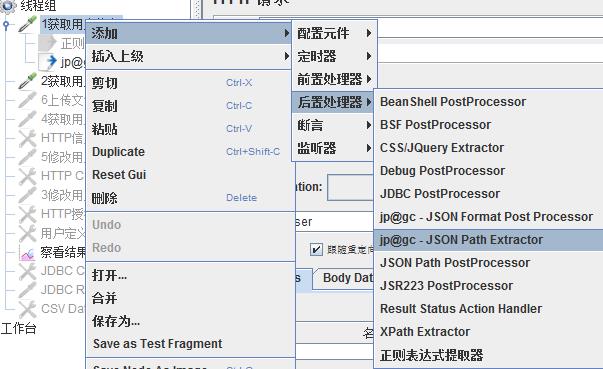
用${age}入参给另一个请求，查看结果传参成功。



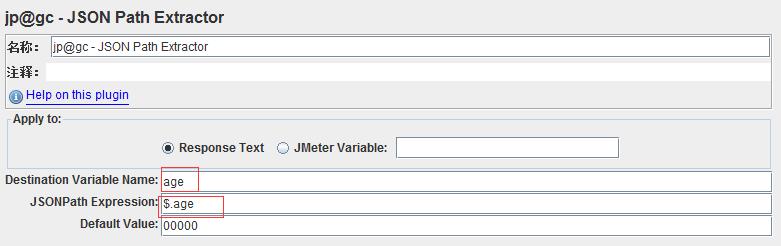


* 用Json Path Extractor 插件关联

只针对Json格式的Response 取值，右击请求添加->后置处理器->JSON Path Extractor；



在JSON Path Extractor中填入引用名和JSONPath Expression；

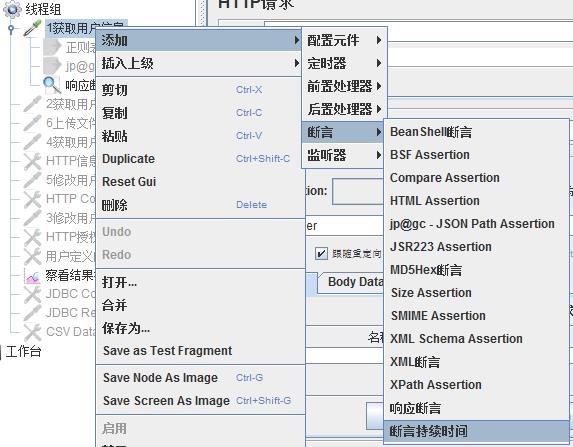


在另一个请求中入参，也和上面的结果一样转参成功。

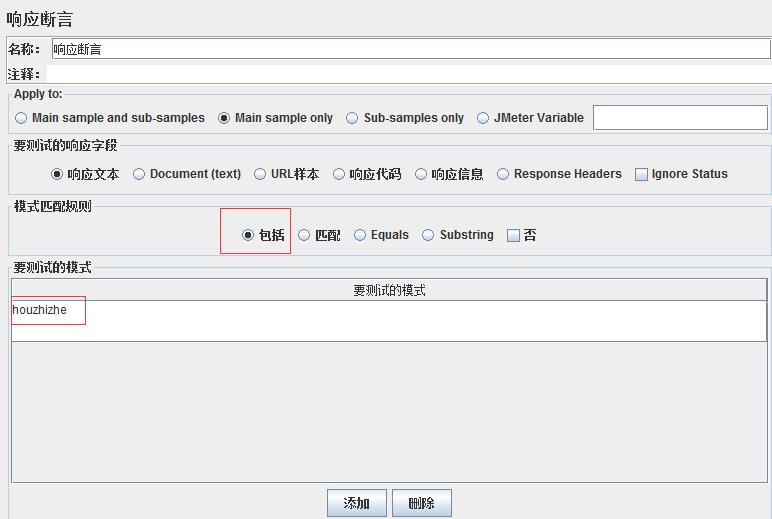
注意，JSON Path Extractor这个插件只能在Jmeter3.x以上的版本安装。

**Jmeter添加断言检查结果**

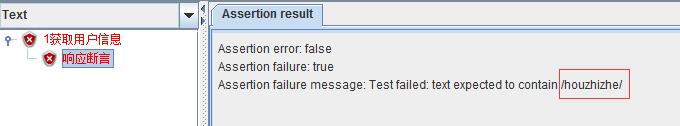
右击请求添加->断言->响应断言；

****

添加并写入希望在响应结果中看到的内容；



发送请求，查看结果；

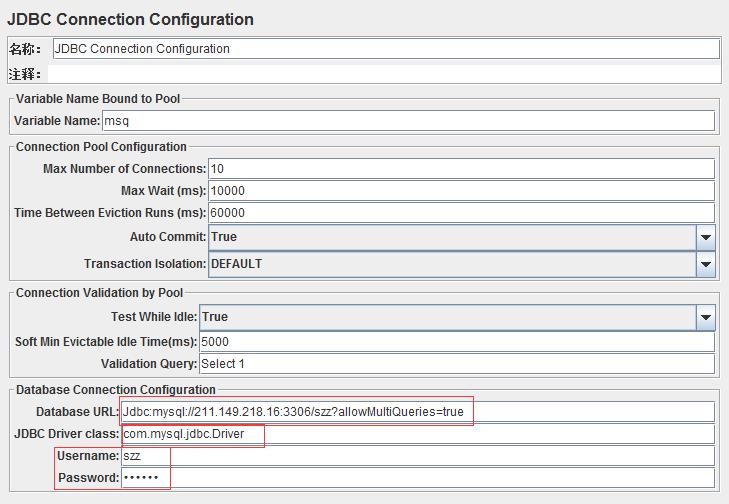


**Jmeter操作Mysql**

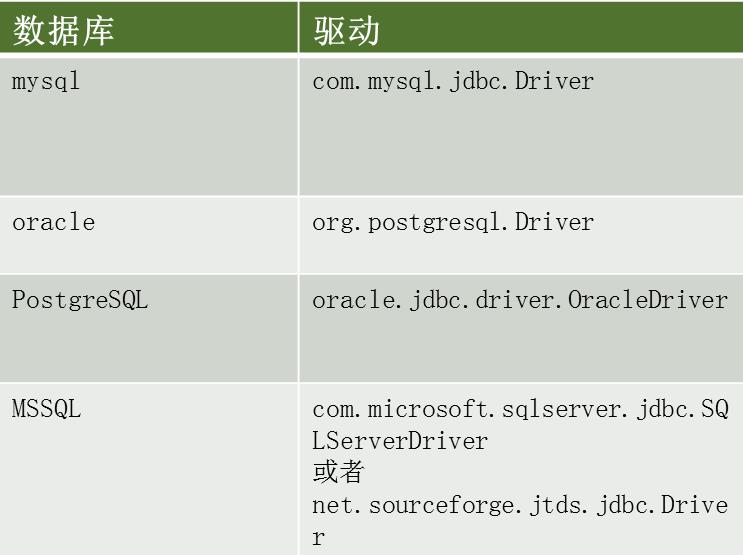
测试计划添加.jar包 mysql-connector-java-5.1.7-bin.jar用于使Jmeter可以读取Mysql；



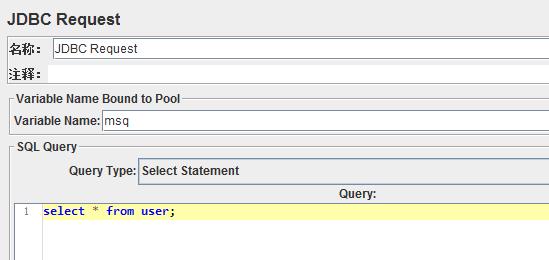
 线程组添加 JDBC Connection Configuration （添加->配置元件-> JDBC Connection Configuration）用于配置Jmeter和Mysql的连接；



JDBC Driver class是固定的，Jmeter操作各类数据的Driver参考下图；



线程组添加JDBC Request（添加->Sampler-> JDBC Request） 用于写sql来操作Mysql；



发送请求，成功访问Msql。



2020非常全的接口测试面试题及参考答案-软件测试工程师没有碰到算我输！

置顶 测试逍遥子 2020-04-08 17:29:39 11327 收藏 661

分类专栏： 接口自动化测试 文章标签： 接口 jmeter 面试

版权

一、前言

接口测试最近几年被炒的火热了，越来越多的测试同行意识到接口测试的重要性。接口测试为什么会如此重要呢？ 主要是平常的功能点点点，大家水平都一样，是个人都能点，面试时候如果问你平常在公司怎么测试的，你除了说点点点，还能说什么呢，无非就是这个项目点完了点那个项目， 这就是为什么各行各业的只要手指能点得动的人都来转行软件测试了。面试的时候面试官希望你除了点点点，还能更深入一点的思考页面上看不到的功能，也就是接口测试了。接口测试可不是下图的USB接口哈，接口是看不见的，但是可以访问！

二、2020非常全的接口测试面试题及参考答案

1、你们公司的接口测试流程是怎样的？（有没有感觉熟悉，貌似在哪里听过）

接口测试我们是在XX项目做的，主要有XX接口，XX接口，XX接口等。

1、首先是从开发那里拿到API接口文档，了解接口业务、包括接口地址、请求方式，入参、出参，token鉴权，返回格式等信息。

2、然后使用Postman或Jmeter工具执行接口测试，一般使用Jmeter的步骤是这样的：

     1、首先新建一个线程组。

     2、然后就是新建一个HTTP请求默认值。（输入接口服务器IP和端口）

     3、再新建很多HTTP请求，一个请求一个用例。（输入接口路径，访问方式，参数等。）

     4、然后创建断言和查看结果树。

3、最后调试并执行用例，最后编写接口测试报告

4、其实我们做接口的时候也碰到了蛮多的问题，都是自己独立解决的，比如返回值乱码（修改jmeter的配置文件为UTF-8编码方式），比如需要登录后才能取得token鉴权码并且这个鉴权码在下面的请求中需要用到（使用正则表达式提取器提取token的值等。

2、请简述一下cookie、session以及token的区别（有没有感觉整个是万年不变的面试题）

（1）cookie数据存放在客户的浏览器上，session数据放在服务器上。而token是接口测试时的鉴权码，一般情况下登陆后才可以获取到token，然后在每次请求接口时需要带上token参数。

（2）cookie不是很安全，别人可以分析存放在本地的cookie并进行cookie欺骗，考虑到安全应当使用session，session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，考虑到减轻服务器性能方面应当使用cookie

（3）可以将登陆信息等重要信息存放为session；其他信息需要保存，可以放在cookie

3、接口测试如何设计测试用例？（必问，有没有感觉答得整个人都不好了？）

接口测试一般考虑入参形式的变化和接口的业务逻辑，一般设计接口测试用例采用等价类、边界值、场景法居多！

接口测试设计测试用例的思路如下：

1.接口业务逻辑测试?(正例)

    接口逻辑测试是指根据业务逻辑、输入参数、输出值的描述，对正常输入情况下所得的输出值

    是否正确的测试，也就是测试对外提供的接口服务是否正常工作。

2.模块接口测试?(反例)

    模块接口测试是为了保证数据的安全及程序在异常情况下的逻辑的正确性而进行的测试。?

    模块接口测试的主要包括以下几个方面：?

    1）鉴权码token异常(鉴权码为空<没有鉴权码>,错误的鉴权码,过期的鉴权码)。

    2）其他参数异常。

        1、必填项检查

        2、参数的长度、类型、格式异常：

                常规参数：(数字、字符串、日期)

                参数长度：6-18位。或身份证、电话的长度。

                参数类型：数字(精度)，字母，中文，带空格的参数，特殊字符。

                日期格式：日期：年月日，年月日时分秒，日期格式(包括/,-,:等)。

    3）错误码异常覆盖。

    4）接口测试其他的关注点

        接口有翻页时，页码与页数的异常值测试

        数据库的增删改查，比如一个post接口操作完成后，通过列表页接口看下新的数据是否和刚才的post一致

        接口返回的图片地址能否打开，图片尺寸是否符合需求

        当输出参数有联动性时，需要校验返回两参数的实际结果是否都符合需求。

        所有列表页接口必须考虑排序值

        所有功能都要考虑兼容旧版本

4、接口测试执行中比对数据库吗？

肯定啊，因为接口返回值的数据来源于数据库，接口对数据的操作还要进行深层次的数据库检查！

5、谈谈你对HTTP协议的了解？

超文本传输协议，端口为80，特点（无记忆功能、快速）是由请求和响应两部分组成请求由请求头、请求行、请求正文组成；响应是由响应头、响应行、响应正文组成，之前我们公司的接口是采用https协议的。

https http+ssl协议 端口443 面向安全的超文本传输协议

6、get和post请求有什么区别？

get和post请求都是客户端向服务器提交的一种请求方式

get是明文传输参数、倾向于请求服务器资源。比如打开网站

post传输数据不可见，安全性高，倾向于向服务器提交数据，比如注册等

7、响应状态码有哪些？

1xx：指示信息--表示请求已接受，继续处理

2xx：成功--表示请求已被成功接收、理解、接受

3xx：重定向--要完成请求必须进行更进一步的操作

4xx：客户端错误--请求有语法错误或请求无法实现

5xx：服务器端错误--服务器未能实现合法的请求

8、在接口测试过程中发现的bug多不多？能举几个栗子？

这个问题其实回到起来很简单，只要做过接口测试的，总能发现几个BUG吧，把你平常发现的bug说2-3个就可以了。 面试官出这个题，主要是想知道你是不是真的做过接口测试，毕竟现在很多小伙伴简历都是写的假的（你要不写估计面试机会都没有，没办法，为了生存，能理解） 比如，提现输入框，在页面上输入负数，肯定是无法提交过去（前端页面会判断金额），如果我不走前端，直接用接口工具发请求，输入一个负数过去。 （假设服务端没做提现金额数据判断） 余额=当前余额（100）-提现金额（-100），那么提现-100，余额就变成200了，也就是越提现，余额越大了。

9、接口测试中有哪些要注意的测试点？

11.1)接口中返回了图片地址，要手工去进行图片的测试（大小、内容）

11.2)接口完成查询功能的时候，数据返回的排序显示

11.3)接口测试的时候，关注参数的默认值、必填项

10、接口执行测试后返回结果做对比，一般比对哪部分内容？

之前必须要对比的就是返回状态码，其次再去对比返回其它关键内容

11、为什么开展接口测试？

13.1)接口测试属于集成测试、测试介入越早、就越能在项目早期发现问题，其修复问题的成本越低

13.2)接口测试非常快速、UI自动化执行一个测试用例10S左右、接口测试用例执行的话，需要的时间是毫秒级的

12、之前在接口测试过程中，使用的工具是什么？

postman或jmeter(5.1)

13、之前用过抓包工具没有？如何使用的？

之前在项目中用过fiddler抓包工具进行HTTP协议请求的抓取

打开fiddler之后，默认浏览器配置了127.0.0.1  8888端口的代理，在fiddler设置好过滤策略后，打开需要进行抓包的网站进行操作，就可以进行抓包

14、json数据是什么，你平时如何解析json数据？

一种开发常用的数据报文格式，由键值对和数组两种格式构成。可以通过工具bejson网站等

15、postman你在工作中使用流程是什么样的？

1) 编写好用例

2) 在postman先建好url环境变量

3) 根据接口用例所属的模块新建集合管理

4) 在集合中不同模块下录入测试用例

5) 录入测试用例的时候根据预期结果在tests页签中增加断言

6) 导出通过Newman去运行

16、postman中设置环境变量有什么用？

在之前项目中，接口测试测试的环境有开发环境，测试环境等，为了测试的时候方便，就在postman设置环境变量，到时所有接口都引用该环境变量，这样就不用为了切换环境导致每次都去修改被测系统接口的主机地址；点击右上角环境变量管理按钮-新建环境变量，在脚本中使用{{变量名}}去调用

17、在接口测试中关联是什么含义？如何用postman设置关联？

关联就是把上一个接口返回值的部分截取出来，作为下一个接口的参数，能让接口串联运行

在postman中设置关联的步骤如下：

1) 先通过正则表达式提取的方式或json取值的方式把下一个接口需要的信息从上一个接口截取出来

2) 使用设置全局变量的代码把取出来的值保存到全局变量

3) 在下一个接口中，使用{{全局变量}}代替要替换的静态值

18、postman支持什么类型的协议测试？

http和https协议的

19、postman参数化有哪几种方式？

内建变量、pre-scripts编写js脚本、批量运行时导入csv或json格式的文件

20、Newman如何执行postman脚本？

Newman run 脚本名称 也可以添加参数生成html报表等

21、jmeter之前用的是什么版本？如何安装的？

jmeter用的是5.1.1版本，安装如下：

先在电脑上安装jdk1.8或以上的版本，然后从官网下载最新的安装包，解压后，进行环境变量的配置，配置好后即安装完成

22、在项目中如何用jmeter进行接口测试？

1) 把线程组数量设置为1，循环次数设置为1

2) 配置好全局变量URL通过配置元件---用户自定义的变量添加

3) 增加配置元件http请求默认值，放置在用户定义的变量之后

4) 添加事务控制器管理和组织测试用例

5) 在事务控制中添加http请求添加测试用例中的接口请求信息

6) 添加对应的断言元件进行断言

23、jmeter中如何设置断言？

右击请求---断言---响应断言---响应断言界面输入要检查比对的项，设置好断言后，执行接口测试如果是通过的，查看结果树不会有任何提示，如果断言失败，就会有红色报错。如果接口返回的数据是json数据，也可以添加json断言

24、jmeter中如何实现关联？

先从上一个接口中通过正则表达式提取器或jsonpath解析器截取下一个接口需要的参数值保存到变量，然后在写一个接口中通过${变量名}去获取

25、jmeter添加http请求默认值元件有什么作用？

添加并设置好后，相当于给所有的http请求取样器都设置了默认值，既不用填写取样器中的比如主机地址、端口、代理等，都可以使用http请求默认值设置的

26、jmeter参数化的方式有哪几种可以实现？

1) 配置元件---用户定义的变量元件可以设置全局变量

2) 函数助手对话框中可以选择比如随机字符串、随机日期、随机数字作为参数化

3) 可以使用csv文件作为参数化，通过配置元件中的csv data set config元件进行设置即可

27. 总结

如果你觉得此文对你有帮助，如果你对此文有任何疑问，如果你对软件测试、接口测试、自动化测试、面试经验交流感兴趣欢迎加入软件测试技术群：695458161，群里发放的免费资料都是笔者十多年测试生涯的精华。还有同行大神一起交流技术哦。

作者：诸葛

出处：https://blog.csdn.net/pengjiangchun

原创不易，欢迎转载，但未经作者同意请保留此段声明，并在文章页面明显位置给出原文链接。

————————————————

版权声明：本文为CSDN博主「测试逍遥子」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/pengjiangchun/article/details/105392005>

接口测试面试题整理

我冷漠 2020-04-27 23:06:38 458 收藏 20

版权

1.什么是接口测试？

接口测试：是测试系统组件间接口的一种测试方法

接口测试的重点：检查数据的交换，数据传递的正确性，以及接口间的逻辑依赖关系

接口测试的意义：在软件开发的同时实现并行测试，减少页面层测试的深度，缩短整个项目的测试周期

2.接口自动化测试的流程？

基本的接口功能自动化测试流程为：需求分析-->用例设计-->脚本开发-->测试执行-->结果分析

3.GET请求和POST请求区别是什么？

GET请求：用于信息获取，相对而言是安全和幂等的；在做数据查询时，建议用GET方式，如：商品信息接口、搜索接口、博客访客接口...

POST请求：表示可能会修改服务器上资源的请求；在做数据添加、修改时，建议用POST方式。如：上传图片接口、登录注册接口...

误区：

“GET是从服务器上获取数据，POST是向服务器传送数据”（该说法有误）

解析：

GET/POST都可以提交数据，GET请求也可以向服务器传递数据，POST请求也需要服务器返回数据

4.接口测试的常用工具有哪些？

Postman

JMeter

SoapUI

5.HTTP接口的请求参数类型有哪些？

查询字符串参数（Query String Parameters参数）一般用于GET请求，会以url string的形式进行传递

请求体参数（Request Body）一般用于POST请求，可以使用Content-Type来指定不同参数类型

6.如何从上一个接口获取相关的响应数据传递到下一个接口？

先从上一个接口中的响应数据获取对应的返回值，然后使用正则表达式or使用JSON解析来提取需要获取的值，然后存储在一个变量中，最后在下一个接口中直接引用该变量即可

7.接口测试用例的编写要点有哪些？

1）必填字段：请求参数必填项、可选项

2）合法性：输入输出合法、非法参数

3）边界：请求参数边界值等

4）容错能力：大容量数据、频繁请求、重复请求（如：订单）、异常网络等的处理

5）响应数据校验：断言、数据提取传递到下一级接口...

6）逻辑校验：如两个请求的接口有严格的先后顺序，需要测试调转顺序的情况

7）性能：对接口模拟并发测试，逐步加压，分析瓶颈点

8）安全性：构造恶意的字符请求，如：SQL注入、XSS、敏感信息、业务逻辑（如：跳过某些关键步骤；未经验证操纵敏感数据）

8.接口测试的步骤有哪些？

1）发送接口请求

2）测试接口获取返回值

3）断言：判断实际结果是否符合预期

9.接口测试中依赖登录状态的接口如何测试？

依赖登最状态的接口，本质上是在每次发送请求时需要带上存储有账户有效信息的Session或Cookie才能发送成功，在构建POST请求时添加必要的Session或Cookie

10.依赖于第三方数据的接口如何进行测试？

可以利用一些MOCK工具（如：JSON Server、Easy Mock）来模拟第三方的数据返回，最大限度的降低对第三方数据接口的依赖

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

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

面试题1：你平常做接口测试的过程中发现过哪些bug?

这个问题其实回到起来很简单，只要做过接口测试的，总能发现几个BUG吧，把你平常发现的bug说2-3个就可以了。

面试官出这个题，主要是想知道你是不是真的做过接口测试，毕竟现在很多小伙伴简历都是写的假的（你要不写估计面试机会都没有，没办法，为了生存，能理解）

比如上面说的，提现输入框，在页面上输入负数，肯定是无法提交过去（前端页面会判断金额），如果我不走前端，直接用接口工具发请求，输入一个负数过去。

（假设服务端没做提现金额数据判断）

余额=当前余额（100）-提现金额（-100），那么提现-100，余额就变成200了，也就是越提现，余额越大了

可以用接口工具去直接请求接口，也可以fiddler抓包，抓到接口后修改金额为负数

所以，接口测试的必要性就体现出来了：

1.可以发现很多在页面上操作发现不了的bug

2.检查系统的异常处理能力

3.检查系统的安全性、稳定性

4.前端随便变，接口测好了，后端不用变

5.可以测试并发情况，一个账号，同时（大于2个请求）对最后一个商品下单，或不同账号，对最后一个商品下单

6.可以修改请求参数，突破前端页面输入限制（如金额）

面试题2：平常你是怎么测试接口的？

通过性验证：首先肯定要保证这个接口功能是好使的，也就是正常的通过性测试，按照接口文档上的参数，正常传入，是否可以返回正确的结果。

参数组合：现在有一个操作商品的接口，有个字段type，传1的时候代表修改商品，商品id、商品名称、价格有一个是必传的，type传2的时候是删除商品，

商品id是必传的，这样的，就要测参数组合了，type传1的时候，只传商品名称能不能修改成功，id、名称、价格都传的时候能不能修改成功。

接口安全：

1、绕过验证，比如说购买了一个商品，它的价格是300元，那我在提交订单时候，我把这个商品的价格改成3元，后端有没有做验证，更狠点，我把钱改成-3，是不是我的余额还要增加？

2、绕过身份授权，比如说修改商品信息接口，那必须得是卖家才能修改，那我传一个普通用户，能不能修改成功，我传一个其他的卖家能不能修改成功

3、参数是否加密，比如说我登陆的接口，用户名和密码是不是加密，如果不加密的话，别人拦截到你的请求，就能获取到你的信息了，加密规则是否容易破解。

4、密码安全规则，密码的复杂程度校验

异常验证：

　　所谓异常验证，也就是我不按照你接口文档上的要求输入参数，来验证接口对异常情况的校验。比如说必填的参数不填，输入整数类型的，传入字符串类型，长度是10的，传11，总之就是你说怎么来，我就不怎么来，其实也就这三种，必传非必传、参数类型、入参长度。

性能测试

接口并发情况，如上面提到的：一个账号，同时（大于2个请求）对最后一个商品下单，或不同账号，对最后一个商品下单

接口响应时间，响应时间太长了，肯定需要优化，一般都是毫秒级别.

面试题3：平常用什么工具测接口的？

接口测试工具很多，首先postman

其次用jmeter

面试题4：webService接口是如何测试的？

webService接口用SoapUI

面试题5：没有接口文档，如果做接口测试？（这是个送命题）

没有接口文档，那还能咋办，瞎测呗！一个公司的开发流程里面，如果接口文档都没有，是无法展开接口测试的，你都不知道这个接口干什么的，也不知道具体每个字段代表什么意思，那还测啥呢？

–当然，你肯定不能回答面试官不测(心理mmp，脸上笑嘻嘻)，接下来就是扯犊子时间

1.没有接口文档，那就需要先跟开发沟通，然后整理接口文档（本来是开发写的，没办法，为了唬住面试官，先说自己整理了）

2.没有接口文档，可以抓包看接口请求参数，然后不懂的跟开发沟通

本题主要靠情商，通俗来说就是忽悠能力，先唬住面试官了再说，进去了也是瞎测测，随时做好背锅的准备。

面试题6：在手工接口测试或者自动化接口测试的过程中，上下游接口有数据依赖如何处理？

用一个全局变量来处理依赖的数据，比如登录后返回token,其它接口都需要这个token,那就用全局变量来传token参数。

面试题7：依赖于第三方数据的接口如何进行测试？

这个标准答案是：mock

接着面试官会问你，如果mock的，然后你就顺着坑继续挖，搭建mock服务，参考这篇【https://www.cnblogs.com/yoyoketang/p/9348552.html】

面试题8：当一个接口出现异常时候，你是如何分析异常的？（抓包）

1.抓包，用fiddler工具抓包，或者浏览器上f12,app上的话，那就用fiddler设置代理，去看请求报文和返回报文了

2.查看后端日志，xhell连上服务器，查看日志。

面试题9：如何模拟弱网测试？

fiddler和charles都可以模拟弱网测试，平常说的模拟丢包，也是模拟弱网测试。

面试题10：如何分析一个bug是前端还是后端的？

平常提bug的时候，前端开发和后端开发总是扯皮，不承认是对方的bug

这种情况很容易判断，先抓包看请求报文，对着接口文档，看请求报文有没问题，有问题就是前端发的数据不对

请求报文没问题，那就看返回报文，返回的数据不对，那就是后端开发的问题咯。

————————————————

版权声明：本文为CSDN博主「我冷漠」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_43748615/article/details/105802166>

软件测试面试题整理(四)之接口测试篇

置顶 JSon liu 2020-11-04 22:28:55 1198 收藏 26

分类专栏： # 软件测试工程师面试合集 文章标签： python 软件测试

版权

https://blog.csdn.net/weixin\_45912307/article/details/109481695

1. 接口测试怎么做的？

原来我们的接口测试，开发这边会给我们一个接口文档，我们根据接口文档编写测试用例，考虑接口正常场景跟异常场景。测试用例编写完成后，我们会用python+request去执行，查看返回的结果是否跟用例的一致，不一致有bug。

需要注意的是参数之间是不是组合关系，如果是组合关系就需要同时考虑，如果不是组合关系就要单独考虑。

也需要考虑正常和异常的场景，多一个参数和少一个参数以及参数为空的情况。

比如我们用python+request做的一个前端注册接口，首先我们把每条用例定义成一个函数模块，再把需要组合的参数编写进用例里面。

比如用户名跟密码这是组合关系就要同时对这两个参数进行考虑。

当用户名与密码正确的情况下，返回的是一个正确的注册成功的信息。

当用户名为空和两个密码一至的情况下返回的提示应该是一个用户名不能为空的信息。

当用户名正确和两个密码不一至的情况下返回的提示应该是一个密码不一至的信息，

当用户名正确和两个密码都为空的情况下返回的提示应该是一个密码不能为空或密码过弱的信息。

当用户名重复，手机号码重复，

勾选了协议和没勾选了协议的情况，还有同时我们也需要用等价类边界值对参数的长度、特殊符号的输入和类型进行考虑。

最后加assert来断言判断返回的参数是否正确。

接口测试的主要好处：在后端测试接口，覆盖更加全面，测试时间可以提前，

原来在接口测试过程中，发现问题最多的，来源于格式的校验这块

比方说，原来我们申请借款，利率设置成5000，也可以申请成功，功能测试，界面已经做了校验，只能在接口中才能测试出这个问题，还有原来我们申请借款时候，需要勾选，同意，协议，接口中不没有勾选，也表示成功。

2. 接口自动化测试怎么测？

原来我们也做了很多接口自动化。接口自动化这块，其实原来我们也是用jmeter请求去做的，这个时候，我们也用到一些工具，http代理，主要方便编写接口请求，通过录制就行了，我觉得接口自动化只是在接口测试中多加了一些参数化、关联、断言参数，主要是函数参数化，自定义变量参数化，文件参数化，主要文件类型csv跟txt，不过原来csv文件用的比较多，还有一些数据库的参数化，断言，主要响应代码断言，响应文档断言。

比方说，原来我们一个登录接口主要是正常场景，异常场景这块。

正常场景：主要是用户跟密码正确，采用数据参数化，把用户名用随机函数进行参数化，随机长度大一些，用户名不存在的情况，原来是通过文件参数化，设置参数值，密码不正确也是通过文件参数化，接口请求中host地址，目录地址，我们都进行数据化，自定义变量去操作，结果检查中，我们主要是用断言来检查，每个请求，设置了2个断言，一个响应代码断言，一般是200，响应文本断言，登录成功，返回码为1，状态提示成功，检查是否成功，

异常场景：都需要设置断言，去检查结果原来做的申请借款接口，需要登录接口cookie，我需要建立2个接口，一个登录接口，一个申请借款接口，通过正则表达式去提取 登录接口返回cookie，在申请借款请求接口，设置http cookie时，值为登录接口返回cookie，还有也要考虑原来我们项目，还有token值，登录返回token提取，当成申请借款的请求参数，当测试场景的脚本编写完成，执行接口测试用例，我们在察看结果树中，检查，主要是看颜色这块，红色检查哪些地方失败，绿色表示通过

3. 你们是怎么做接口自动化的

原来我们接口 主要是用的python+requests去运行的

首先，开发会给我们一个接口文档，拿到接口文档后，我们就进行测试点的分析，考虑正确场景，条件的组合，异常场景，多一个参数，少一个参数，参数为空的情况，

比如原来我们 做一个生成订单的接口，考虑正常场景，异常场景正常场景就是不同的 订单类型，订单金额，能不能申请订单，每个参数的格式类型的校验，异常场景，多一个参数，少一个必填参数的时候，还有参数为空的情况，原来我们是用python+request去做的接口。

首先，导入request包。建立一个 headers，保存请求头的信息，因为订单请求方式 是post类型，数据格式是 form表单格式，我们把数据保存到data的字典里面，这个时候我们还需要 登录的cookie值跟登录后产生的token值，我们会去通过动态关联去获取 登录的token跟cookies，cookies值的话，我们是直接调用登录返回的 cookies，token值的时候，我能是通过导入re模块，通过正则表达式去提取当参数，headers，cookies输入完成以后，我们就发送请求，打印返回结果，检查返回结果是否跟我们测试用例一致当运行其他测试用例时，我们去修改data里面的参数就行，在发送请求，有的请求时https协议的时候，我们发送请求的时候 还会verify=false 去忽略掉证书验证对应多个接口调用 cookies 我们会用到session去保存，接口发现比较多的问题，就是格式校验这块。比如说我们提交订单，订单数据没有显示，订单格式也没有显示，输入字母，汉字都可以订单类型 为空，也会生成订单成功，我觉得接口可以发现接口更多的bug，还可以提早进行测试，提高测试的质量。

4. 接口自动化

原来我们接口自动化是用 python+request+unittest执行。接口自动化其实主要就是接口测试的基础上填加了断言、参数化、动态关联，做接口自动化之前，我们也会划分模块：报告、公共的模块、测试数据、测试报告，主要的目的是为了方便后期的维护。

测试数据，一般原来我们就是用的接口测试用例；

公共的模块，主要是里面的一些公共的操作，比如说用例excel数据的读取，数据库的连接、还有我们封装的每个接口请求;

断言的主要是 获取访问接口的值判断，用的是assert;

参数化主要用的比较多是excel表格，就是测试用例数据，

还有需要调用登录后的cookies跟token的时候，我们就会用到关联;

比如说原来我们写的一个 申请借款的接口吧。

首先我们会编写测试用例，把每个用例数据保存到excel中再建立一个 申请借款的模块，这个时候我们去调用申请借款的功能模块，里面的参数我们是保存在excel表格中，我们建立发送请求，通过参数化，去读写excel表格中的数据，获取到返回的数据，通过assert 去对应返回的数据跟用例中异常的数据，这个时候也会做数据库断言，去连接数据库去查询数据库中时候存在查询，如果是返回结果是json数据格式，我们还会转化下格式后，再去断言，这个申请借款模块，也会用到登录的cookie值token，我们先建立一个登录的请求，提取返回的cookie值token，excel表格多个用例，我们就用到循环去运行，读取excel中用例总的条数，去循环运行，这里要注意的是，就是excel表格数据时是str我们要eval转化成字典格式，把每个接口封装好以后，我们就会调用unittest框架去运行所有test文件的测试用例，如果只是执行部分用例，也可以通过unittest框架来指定接口自动化

我个人觉得，性价比是比较高的，实现起来简单、维护成本低，容易提高覆盖率等特点。接口是稳定的，最多是增加一个字段或者新增接口之类的低成本，有了相对的稳定性，不需要大量重新编写脚本，只需要基础维护包括用例的扩充就可以了，执行的快，反馈的速度快

5. 平常你是怎么测试接口的？

通过性验证： 首先肯定要保证这个接口功能是好使的，也就是正常的通过性测试，按照接口文档上的参数，正常传入，是否可以返回正确的结果。

参数组合： 现在有一个操作商品的接口，有个字段type，传1的时候代表修改商品，商品id、商品名称、价格有一个是必传的，type传2的时候是删除商品，

商品id是必传的，这样的，就要测参数组合了，type传1的时候，只传商品名称能不能修改成功，id、名称、价格都传的时候能不能修改成功。

接口安全：

1、绕过验证，比如说购买了一个商品，它的价格是300元，那我在提交订单时候，我把这个商品的价格改成3元，后端有没有做验证，更狠点，我把钱改成-3，是不是我的余额还要增加？

2、绕过身份授权，比如说修改商品信息接口，那必须得是卖家才能修改，那我传一个普通用户，能不能修改成功，我传一个其他的卖家能不能修改成功

3、参数是否加密，比如说我登陆的接口，用户名和密码是不是加密，如果不加密的话，别人拦截到你的请求，就能获取到你的信息了，加密规则是否容易破解。

4、密码安全规则，密码的复杂程度校验

异常验证：

所谓异常验证，也就是我不按照你接口文档上的要求输入参数，来验证接口对异常情况的校验。比如说必填的参数不填，输入整数类型的，传入字符串类型，长度是10的，传11，总之就是你说怎么来，我就不怎么来，其实也就这三种，必传非必传、参数类型、入参长度。

性能测试

接口并发情况，如上面提到的：一个账号，同时（大于2个请求）对最后一个商品下单，或不同账号，对最后一个商品下单接口响应时间，响应时间太长了，肯定需要优化，一般都是毫秒级别

6. 接口测试经常遇到的bug和问题，如下：

（1）传入参数处理不当，导致程序crash；

（2）类型溢出，导致数据读出和写入不一致；

（3）因对象权限未进行校验，可以访问其他用户敏感信息；

（4）状态处理不当，导致逻辑出现错乱；

（5）逻辑校验不完善，可利用漏洞获取非正当利益等。

7. 接口自动化测试中携带哪些参数

Url： http、https、tcp、ftp

Method： post、get、delete、put

Data： application/json、application/x-www-form-urlencoded、multipart/form-data、text/xml

Headers： content-type、user-agent、cookie、token、authorization

8. 当一个接口出现异常时候，你是如何分析异常的？

答：先抓包，用 fiddler（charles）工具抓包，或者浏览器上 F12 调试工具；APP 上的话，那就 用 Fiddler 做代理，通过手机设置代理去看请求和返回报文； 查看后端日志，如 Linux 系统通过 xhell 连上服务器，查看接口日志，查看是否有报错信息 （命令：tail -f 日志文件）；

9. 接口测试中，依赖登录状态的接口如何测试？

答：依赖登录状态的接口的本质上是在每次发送请求时需要带上 session 或者 cookie才能 发送成功，在构建 POST 请求时添加必要的 session 或者 cookie

10. 平常用什么工具测接口的？

答：常用 http 协议接口测试工具，如：postman、fiddler、jmeter；webService接口用SoapUI、 jmeter 等。

11. 没有接口文档，如果做接口测试？

答：用抓包工具把接口抓取处理，然后针对性进行测试；接口中字段信息不清楚的，找时间集中寻求开发解答。（常用抓包工具 Fiddler、Charles 等）

12. 怎么设计接口测试用例？

通常，设计接口测试用例需要考虑以下几个方面：

①是否满足前提条件 :有些接口需要满足前提，才可成功获取数据。常见的，需要登录 Token

逆向用例：针对是否满足前置条件（假设为 n 个条件），设计 0~n 条用例；

②是否携带默认值参数

正向用例：带默认值的参数都不填写、不传参，必填参数都填写正确且存在的“常规”值， 其他不填写，设计 1 条用例；

③业务规则、功能需求 这里根据时间情况，结合接口参数说明，可能需要设计 N 条正向用例和逆向用例；

④参数是否必填 逆向用例：针对每个必填参数，都设计 1 条参数值为空的逆向用例；

⑤参数之间是否存在关联 有些参数彼此之间存在相互制约的关系；

⑥参数数据类型限制 逆向用例：针对每个参数都设计 1 条参数值类型不符的逆向用例；

⑦参数数据类型自身的数据范围值限制 正向用例：针对所有参数，设计 1 条每个参数的参数值在数据范围内为最大值的正向用例；

13. 请问你们公司是如何做接口测试的？

答： 接口测试实际跟一般测试不同就是测试用例的设计部分。

①获取接口规范；

②设计接口测试功能用例（主要从用户角度出发看接口能否实现业务需求）；

③各种入参验证（正常情况，异常情况包括输入参数个数不对，类型不对，可选 / 必选，还有考虑参数有互斥或关联的情况）；

④接口返回值各种验证（符合接口文档需求）；

⑤了解接口实现逻辑，实现逻辑覆盖（语句 / 条件 / 分支 / 判定 /…）；

⑥接口能并发执行吗、安全吗，性能满足要求吗；

⑦采用工具或者自写代码来验证；

⑧发现问题跟功能测试一样，该报bug报bug，该跟踪状态的跟踪状态；

14. 测试接口，接口信息从哪里获取呢？

常用的有三种方式：

1 通过抓包工具比如fiddle，charles获取接口信息

2 通过浏览器开发者工具，networks查看接口请求信息

3 当然最直接和最靠谱的就是接口文档，这就是接口的需求文档。一个规范的接口文档最基本的应该包含了（接口请求地址，请求方法，请求头信息说明，接口入参说明(包括参数的类型、是否必填、长度范围等)，接口响应示例等）。

15. 什么是接口测试？

通过测试不同情况下的输入参数和与之对应的输出结果来判断接口是否符合或满足相应的功能性、安全性要求；

接口测试就是代替前端或者第三方，来验证后端实现是否符合接口规范.；

16. 常见接口

webService接口：是走soap协议通过http传输，请求报文和返回报文都是xml格式的，我们在测试的时候都用通过工具才能进行调用，测试。可以使用的工具有SoapUI、jmeter、loadrunner等；

http api接口：是走http协议，通过路径来区分调用的方法，请求报文都是key-value形式的，返回报文一般都是json字符串格式，有get和post等方法，这也是最常用的两种请求方式。可以使用的工具有postman、RESTClient、jmeter、loadrunner等；

17. 接口测试的好处?(为什么要做接口测试)

测试接口的正确性和稳定性, 能快速定位bug,提高测试效率.

能为项目平台带来高效的缺陷监测和质量监督能力；

平台越复杂，系统越庞大，接口测试的效果越明显（提高测试效率，提升用户体验，降低研发成本）

18. 接口测试原理

模拟客户端向服务器发送请求报文，服务器接收请求报文后对相应的报文做处理并向客户端返回应答，客户端接收响应数据后并进行判断

请求: 是否正确, 默认请求成功是返回200, 假如请求错误返回400, 404, 500等状态码

检查: 返回数据的正确性与完整性

安全性: 接口一般不会暴露在网上任意被调用,需要做一些限制,比如必须登录或者请求次数、频率限制

19. 接口测试流程:

定位服务器接口资源并提交测试数据，然后查看响应结果是否符合预期

定位接口资源(URL)

提交测试数据

检查响应结果

20. post请求体数据类型？



————————————————

版权声明：本文为CSDN博主「JSon liu」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_45912307/article/details/109481695>

接口自动化测试面试题，建议收藏哦！

CC爱测试 2021-03-20 22:07:08 207 收藏 7

分类专栏： 自动化测试 学习 文章标签： python 软件测试

版权

1、get和post的区别？

l http是上层请求协议，主要定义了服务端和客户端的交互规格，底层都是tcp/ip协议

l Get会把参数附在url之后，用？分割，&连接不同参数，Get获取资源，post会把参数放置在http请求体中，Post发送请求数据

l Get产生一个tcp数据包，post产生两个tcp数据包

l Get请求，浏览器会把http header和data一起发送，服务器返回200响应码

l Psot请求，浏览器先发送header，服务器响应100（continue），然后再发送data，服务器返回200响应码

l Post的安全性比get高，如果浏览器缓存了，会把get参数留在缓存中，用户可以看到参数信息

2、post请求的四种参数形式是什么？

Application/x-www-form-urlencoded：将参数以一组键对值的方式发送

Multipart/form-data：支持单组和多组文件上传

Application/json：根据后端接口的定义支持数据类型

Text/xml

3、http状态码302, 403, 503分别代表什么？

302：临时重定向到某一个页面，比如要登录之后才能进入的页面，他首先会临时重定向到登录界面

403：权限不够 服务器理解客服端的请求，但拒绝此请求

503：服务端目前无法使用，过载或者维护中

500：服务端异常

404：请求失败，请求所希望得到的资源未被在服务器上发现

401：请求身份验证

1XX系列：指定客户端应相应的某些动作，代表请求已被接受，需要继续处理

2XX系列：代表请求已成功被服务器接收、理解、并接受。这系列中最常见的有200、201状态码。

3XX系列：代表需要客户端采取进一步的操作才能完成请求，这些状态码用来重定向，后续的请求地址（重定向目标）在本次响应的 Location 域中指明。这系列中最常见的有301、302状态码。

4XX系列：表示请求错误。代表了客户端看起来可能发生了错误，妨碍了服务器的处理。常见有：401、404状态码。

5xx系列：代表了服务器在处理请求的过程中有错误或者异常状态发生，也有可能是服务器意识到以当前的软硬件资源无法完成对请求的处理。常见有500、503状态码。

4、公司有用到第三方服务，出了问题，打电话给第三方，第三方不承认，这时候日志又显示不到错误，应该怎么处理？

1/ 单独调用第三方服务，查看三方接口返回的status，code，msg，看是否能支持正常服务

2/ 在自己的应用中mock第三方服务，来调用自己的接口，看能否正常工作，将结果反馈给第三方

3/ 配置日志级别在debug级别时，添加第三方请求过程的日志，查看入参，响应来追踪问题

5、接口自动化中的关联怎么处理？

把上一个请求返回的结果传入到下一个请求的参数中，将请求的结果反射到一个类属性（使用setattr（）函数），下一个请求去调用这个类属性

6、自动化测试怎么校验结果？

断言 ，预期结果与实际结果对比

数据库校验，根据测试场景来查询数据库里的数据和请求之前的数据进行比对

7、自动化使用的测试框架是什么？简述自动化框架的设计、维护

1/测试框架：python+unittest+requests+ddt+openpyxl+pymysql+logging

测试框架：

python：入门简单，语法简洁

unittest ：定义一个测试用例类，具体的方法来维护测试用例的生命周期，测试场景行为，测试用例 前置场景，行为，期望结果，实际结果，断言方法，Setup teardown方法

requests：接口调用 ，支持http请求的库，API 简洁，提供不同的http请求方法，支持session，cookies，

ddt ：数据驱动，ddt 类装饰器，data 测试方法装饰器 unpack解包可迭代的数据类型

普通用户，数据库，配置文件—（基础数据）

openpyxl： 数据管理 excel管理数据，使用openpyxl模块来进行excel数据的读和写（excle，csv, json, yaml, txt都可以管理测试数据）

pymysql：数据库交互，数据校验

eval，json：数据格式的转换 Eval将python支持的格式转换成对应的格式

logging：日志处理， 统一日志输出格式，渠道，级别，执行结果的记录，便于定位问题

jenkins：持续集成

2/框架设计思路：数据驱动+结构分层（可读性，可维护性，可扩展性）

数据驱动：将维护数据与代码分离，接口调用行为一致，针对不同的参数组合驱动不同的测试场景，减少代码冗余

结构分层：数据层+用例层+逻辑层

数据层：测试数据的支撑 data.xls

用例层：用例的执行 test\_register.py test\_recharge.py

逻辑层：公用的方法的封装与提取 doexcle.py do\_mysql.py http\_requests.py logger.py等模块

3/框架设计步骤：

准备测试数据： EXCEL表准备测试用例—excel数据的读取—参数值的替换

发起请求：请求方法（get/post方法进行封装—URL的拼接（不同—参数转化为字典

拿到请求的返回值：解析返回值code，status，msg信息

断言

好处：

1、自动化测试用例和手工测试用例的完美结合，减少重复工作

2、配置灵活，可以自主切换测试环境，执行测试用例

3、常用功能进行封装，逻辑清晰，易于维护

4、统一执行入口，管理测试用例集：

run.py模块通过模糊查找来选择需要执行的测试用例

5、持续集成，定时构建，快速反馈

8、具体的在这个项目中自动化怎么应用到实际的，您对自动化结果的分析

完成所有的自动化测试框架的设计和实现后，进行接口测试，然后集成到jenkins，配置定时执行，生成html报表，查看测试通过率，查看接口的功能

每次发版时，进行回归测试，新功能开发未提测前

9、做好测试计划的关键是什么？

明确测试目的，增强测试计划的实用性，保证测试用例的实用性和覆盖率，完全需求文档和规格指标书等，严格界定测试周期，测试轮次，测试覆盖模块等

大佬云集自动化测试交流群：1033155635（加群暗号：123）

————————————————

版权声明：本文为CSDN博主「CC爱测试」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/waitingwww/article/details/115035955>