# **[最全最新2020年软件测试面试笔试题目](https://www.cnblogs.com/ITester520/p/12542203.html)**

# 高频问题：

自我介绍：

面试官好，我叫\*\*，一共就职于两家公司，工作已有五年时间，平时主要对web、APP及公众号进行相关测试方面的工作以及分配测试任务，定期与组员分享测试技术。从事软件测试的工作呢，是我一直都感兴趣的，后面也是想能在这个职位上长期发展下去，这两天也了解了一下您们公司及相关的产品，希望能够加入公司从事软件测试这个工作，谢谢；

## 工作中印象深刻的 BUG

我们换位思考一下，面试官问这个问题的目的是什么？

其实，它并不关心你描述的这个bug是否真的有价值，或有多曲折离奇？他只是：

**\* 了解你平时工作中的测试能力**

所以，这就要求的你平时工作中遇到bug时试着自己去定位，定位bug的过程远比你的单纯的执行测试用例有“价值”（自我技能提高的价值），在定位bug的过程中你需要掌握和运用更多知识。

另外，建议你平时养成总结的好习惯，发现的bug，开发解决了，最好问问他原因以及解决的方法，这样再遇到类似问题时，自己也可以试着定位解决。遇到难解决的bug，也可以把最终的解决过程记录下来。（这不是就有素材了）

**举几个例子：**

(1)有的用户在系统成功上传文档之后，点击文档的名称没法进行预览。后面是开发做了优化，发布后用户才能使用了。（这个bug是我第一次真正意识到兼容性测试的重要性，也增加了我对兼容性测试的认识）

(2)我遇到的一个就是：  
需求：做一排单选功能，能实现单选的功能  
开发：做的是点击单选按钮，能选中该内容  
测试：也是这么测的  
  
上线后，领导说难用，很难点中那个小圆点。  
原因：只实现了选中单选小圆圈，并且圆点点很小，有选中的作用，没有做成点击该行，即可选中。这个问题貌似大家都实现了需求，却没有很好的考虑易用性。

## 2、项目中负责的模块？并说一下你设计的用例？(包括功能和接口， 考验基本功)

## 3、怎么设计功能测试用例

**1、显式功能性需求**

对于显式功能性需求我们最长用到的方案主要有三种：

等价类划分法

边界值分析法

错误推断法

**1.1、 等价类划分法**

我们如果想测试一个功能的最傻的办法就是穷举。比如说一个密码验证功能，我们把所有的可能的密码都尝试一遍，自然就可以覆盖掉到所有的问题与可能。但是这种穷举的方法明显是做不到的。因此我们要用到等价类划分法。

等价类划分法就是说我们将所有可能的输入数据或操作分为多组不同的子集，每个子集中的数据与操作对发现程序中的潜在错误都有同等的效应。这样我们就将一个子集称为一个等价类。比如输入各种与用户名不相符的密码，是一个等价类；输入各种不存在的用户名是另一组等价类。这样在测试的时候，我们只要在每个等价类中选择一个典型操作，就可以达到较好的测试覆盖度。

等价类还会分为有效等价类和无效等价类两种。有效等价类指的是合理的、有意义的输入，主要用来验证功能是否实现了某个功能。无效等价类与有效等价类相反，指的是无意义的，超过软件规格的，不合理的输入，主要用来测试功能的健壮性，看是否考虑了如何处理不合理的情况。

**1.2、边界值分析法**

在我们在测试合理与不合理的数据的时候，往往最容易出现问题的就是合理与不合理的边界，这时我们就需要使用边界值分析法了。边界值分析法，就是对恰好大于、小于和等于边界的值进行测试，来验证程序是否做到了合适的处理。边界值分析法一般是作为等价类的补充，来加强测试功能实现的程度与健壮性保障的程度，是否符合规格。

**1.3 错误推测法**

在测试的时候就算我们使用了等价类划分法和边界值分析法，也很可能会遗漏一些需求中没有清晰提出，技术上比较隐蔽的错误。这种错误就需要测试人员通过已有的经验、对功能实现可能的方法的理解或直觉，来推测出软件中可能存在的各种错误与场景，然后编写测试用例来进行验证，这就叫做错误推测法。比如，登录超时后，某个需要权限操作的功能在使用的时候，是否跳到了登录页，还是直接报错，甚至说依旧可以操作。这种错误是需要测试人员一定的经验、技术积累与直觉的。

虽然说功能性测试往往是黑盒测试，但是如果测试工程师对于功能的实现有一定的理解——比如说是否用了缓存、是否使用了消息队列、是否某个地方会消耗极大的性能等等——将会更容易的推断出哪些地方会产生错误。

**2、非功能性需求**

在测试工程师测试完显示功能需求之后，还要考虑到系统的非功能性需求。这种需求可能在文档中有明确提到，也可能并没有明确的提出。但是我们的测试工程师在测试的时候却必须要关注到。

**2.1 兼容性测试**

兼容性指的是开发的软件是否在各种平台都可以使用。比如我们开发一个网站，我们的用户可能会用到各种不同的浏览器访问我们的网站。这样我们在测试的时候，就不能只考虑到某一种浏览器。我们需要考虑到多种浏览器的兼容性。

兼容性测试可能会涉及到：

不同厂商的浏览器及相同厂商不同版本的浏览器。

不同的设备终端及操作系统

不同的屏幕分辨率

不同的用户软件环境（比如是否禁用了cookie、是否可以连接外网等）

**2.2 安全性测试**

我们的测试人员还需要关注到开发软件的安全性。这涉及到：

用户隐私信息是否加密

需要权限的资源是否有没有权限也可以被拿到的风险

会不会受到跨站脚本的攻击

会不会受到sql注入攻击  
等等

**2.3 压力测试**

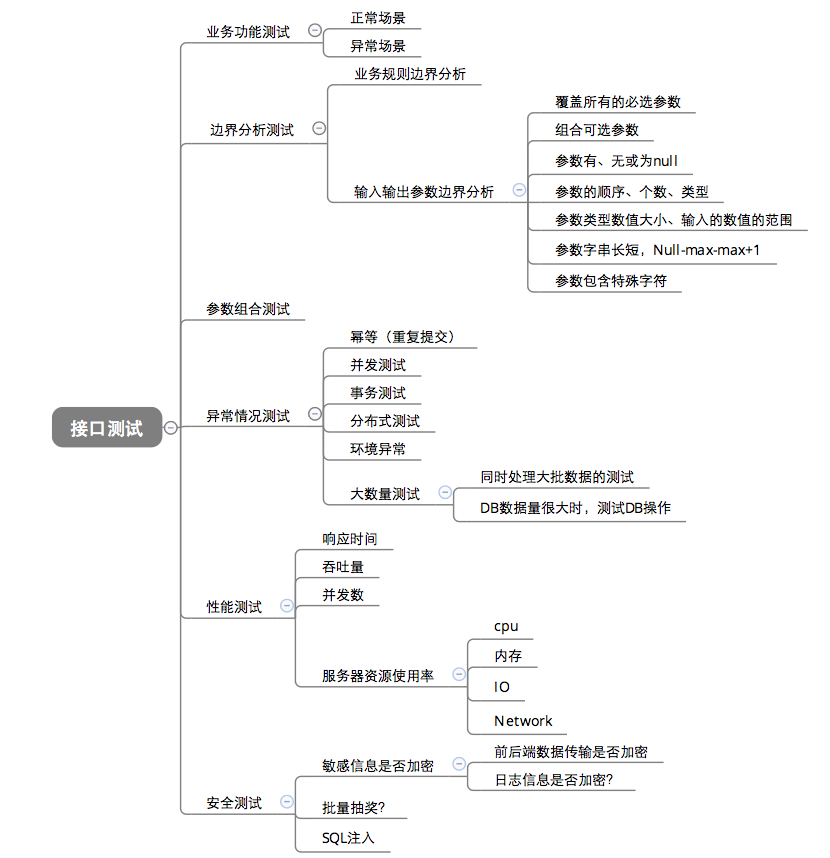
测试人员也需要考虑的软件是否能够承载其需求所需的压力，例如：

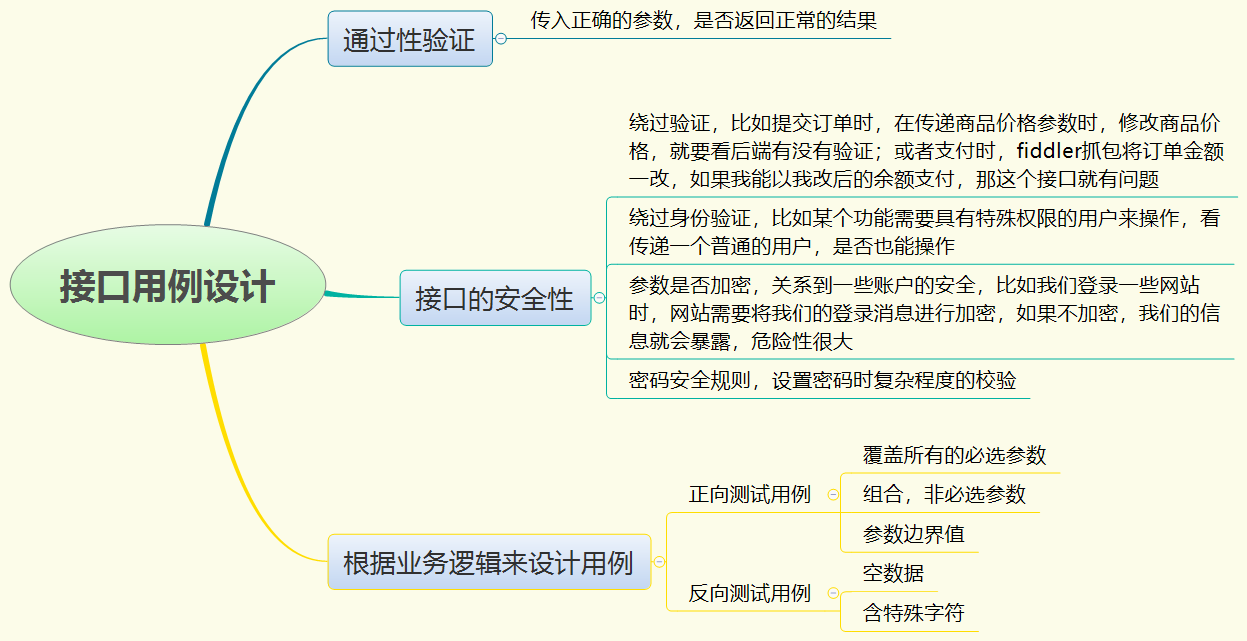
软件是否能在合理的时间内响应用户行为

软件是否可能承载足够的请求

软件在处理大数据量时会不会产生资源锁死  
等等

## 怎么设计接口测试用例





主要从四个方面来设计接口用例：**功能，业务逻辑，异常，安全；**

　　一、**功能：**

　　1）功能是否正常；

　　2）功能是否按照接口文档实现

       举例：比如博客园添加随笔，需要登录才能添加。也就是业务要求不支持游客添加随笔功能，如果设计一个没有登录的用户，然后去测试添加随笔接口，结果接口能添加到随笔，说明功能不正常，不符合需求和接口文档描述。

　　二、**业务逻辑：**是否依赖业务；  
       举例：该接口调用之前，需要调用登录接口，如果不登录也能请求数据，不符合业务逻辑。

　　三、**异常：**参数异常和数据异常  
　　1）参数异常：关键字参数，参数为空，多，少参数，错误参数

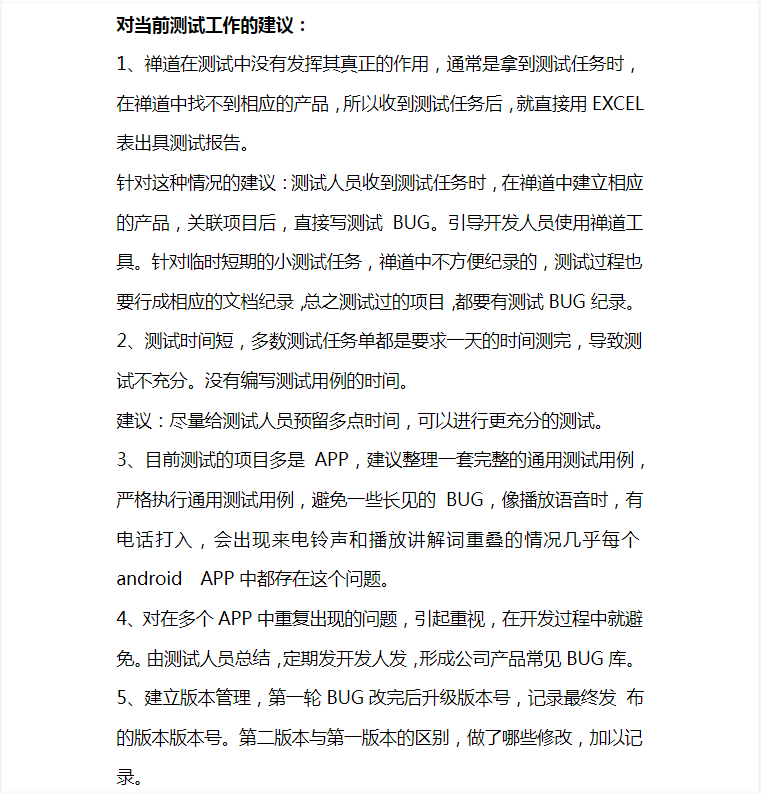
　　2）数据异常：关键字数据，数据为空，长度不一致，错误数据  
       举例：不管数据异常还是参数异常，测试点差不多，一个参数有key和value，key表示参数，value表示数据。第一，看看参数和数据能不能支持关键字，例如Java中的保留关键字等等。第二个就是参数和数据都为空，看看是否做了判断。第三个，参数多和少，例如有两个参数的接口，你需要设计一个三个参数的用例，一个只有一个参数的用例。数据那边长度不一致，例如设计很长的字符串是否支持，因为数据库创建表过程都设置好了每个字段的长度。输入错误的参数和数据，例如故意输出单词等等。

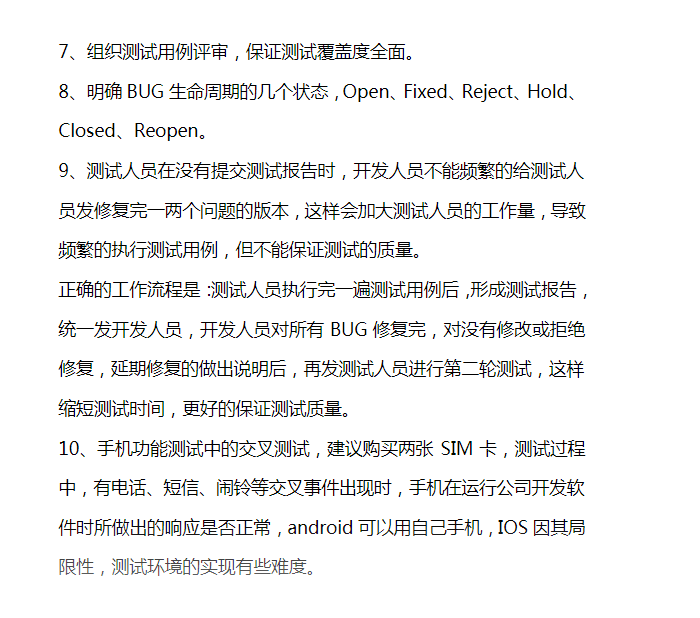
　　四、**安全测试用例设计：**  
　　1)cookie：有cookie才能获取数据，如果不带cookie还有信息返回，说明有问题  
　　2)header：正常接口带header信息，删除header看是否能够返回数据。  
　　3)唯一识别码：app手机识别码，一般是唯一的。

       安全测试主要从上面三点检查。第三个是唯一识别码，主要是指app上手机的识别码，一般很少用到，除非很严格的接口测试，例如银行app登录，需要指纹，而指纹来源手机，一般有一个手机识别码判断过程。

## 对测试流程改进以及测试质量保证提过哪些意见建议？(意思就 是除了大家都能做的这些你对公司还有什么特别的贡献，答好很加分)

1. 测试人员介入太晚，基本上都是等代码开发完成时，才介入。  
     
   改进建议：在项目需求开始启动时，就能有相应测试人员跟进，并随之开始测试的一系列活动；由于在需求阶段可能测试人员的工作量可能较少；可以让其同时兼顾其它项目任务；到测试用例编写阶段才全职投入。  
     
   2、需求变化太多太乱，相关文档没有随之更新，文档与项目实际功能不相符；造成很多时候最新的需求都只是藏在个别人的脑中，而测试人员总是最后一个知道需求变化的人。  
     
   改进建议：能够建立需求变更体系，到什么阶段时必须停止需求变更（必须在项目前期就让需求提出人明确这一点）；每次需求变更必须让需求提出人员确认；需求变化后必须由专人更新相关文档（这些文档都是测试人员编写计划及用例的依据）；并能知会相关人员。这样才能做到程序人员修改相应的程序，测试人员修改相应的用例，且能对需求变更后的程序进行正确的测试。  
     
   3、项目无分阶段送测，造成无法对阶段性的开发成果进行测试，测试任务积压、问题无法及时发现。  
     
   改进建议：在项目前期制定项目计划时，将各阶段性送测纳入到计划中；开发人员根据项目计划进行开发，而测试人员可根据项目计划来安排自身编写用例的先后顺序以及在各阶段性送测时间要求送测（此阶段性送测可视项目的大小、项目与其它系统的关联来定好每个阶段送测的内容）；  
     
   4、测试人员没有独立的稳定的测试环境，无法控制版本更新；造成太多重复无效的测试，也常会因为提交无效的BUG。  
     
   改进建议：最好可以给测试人员提供一个独立的测试环境（包括数据库以及应用程序都是独立的），以保证测试人员所有的BUG都是稳定的、可重复的环境下发现的；如果不能做到独立测试环境情况下，也尽量做到能让测试人员去控制版本更新的频度，以控制测试的有效性。  
     
   5、没有一个行之有效的BUG跟踪机制；造成BUG提交重复、回归不及时，或不能正常被回归。  
     
   改进建议：项目中的所有成员都利用同一种方式去进行BUG提交、跟踪；BUG分发人员需做到BUG的过滤，来达到BUG的有效性；在BUG表单中需要能及时体现BUG的最新状态；这样项目组人员才都能对项目中已发现、已解决的BUG做到心中有数；项目管理人员也可以对整个项目的状态做到心中有数。  
     
   6、无完善的测试用例检查机制，无法对测试过程进行检查，也即无法保证测试结果的真实性。  
     
   改进建议：测试人员之间对测试用例进行走查，或开发人员分配时间出来对测试用例进行走查，以保证用例对需求的覆盖率；测试用例能与BUG对应，使用工具管理用例在每个阶段的执行情况，以对测试过程进行跟踪。  
     
   7、开发人员与专业测试人员比例严重不合理，而非专业测试（用户测试）介入太早。  
     
   改进建议：一般在国内的情况，测试人员与开发人员的比例是1:5左右，但在这里的比例严重不止，一个测试人员都负责多个项目，严重影响测试的质量；对于用户测试一般是在项目验收阶段才介入的，现在都提前介入了；如果用户必须提前介入测试，且作为测试的人力来算的话，那应该由测试来统一安排这些调度，以保证测试的合理分工。





## 怎么开展自动化测试工作的

**一、**

**自动化测试不宜大力投入**

不管是自动化测试还是接口测试，都不宜在前期大量投入。当前业务是否适用自动化，哪些框架或者工具更适合，适合做接口自动化还是 UI 的自动化？如何让自动化达到收益和效果？

这些问题都需要根据团队和业务具体的情况去尝试，找到合适的才行。如果前期投入太大，团队对其期望太高，非常容易在遇到一点挫折的时候对自动化丧失信心。

另外，投入太大，毕竟加大人力或者减少手工测试的投入，会造成测试资源的紧张。在项目时间和成本的压力下，测试管理者需要顶着巨大的压力，这本身就很难成功。

可以安排一些技术强或者技术兴趣浓厚的成员，在项目允许的情况下减少其手工测试工作量，让其可以利用部分工作时间和部分业余时间尝试做自动化，先局部功能尝试，能够有效果，在扩大范围。逐步达到一定的自动化规模。

**以接口为主UI为辅**

UI 自动化因其运行环境的问题，会导致运行速度慢，对环境依赖过高。同时因程序界面的多变性，导致自动化脚本维护成本大大增加。

接口测试有很多优于 UI 自动化的地方。但是接口测试也自有其短板，对流程性质的测试并不适合用接口自动化来覆盖。接口自动化更适合覆盖单一接口功能的检查。

所以我们可以采用核心业务流程使用 UI 自动化，单一功能使用接口自动化，两种层面的自动化结合的方式来进行。

**不轻谈自动化测试平台**

目前测试界开始流行起自己开发测试平台（以接口为主）。简单来说就是模仿常见的接口测试工具，用 Python 或者 Java 写成了一个 web 版本的。

当然也有其理由，“定制性更强！”但是毕竟是软件都需要进行测试，花大量精力开发的平台并不稳定，而本身功能和理念上并没有太多更新。而这样的一些功能，市面上的绝大部分测试工具都能胜任。

如果是为了提高个人能力，项目时间有空余，怎么玩都可以。若是要在实际工作中应用，那么就有点得不偿失了。

自动化测试中，工具的重要性始终是最低的。理念、思维和环境治理才是最重要的。

通过不断小范围试错，找到适合自己团队的自动化策略才是最重要的。任何技术脱离实际应用都是耍流氓。

**二、**

很多时候测试人员在开展自动化测试的时候都是不加思考的就开始做，认为用工具录几个脚本再回放就可以了，没有考虑有效性；而做了并不一定就会有效果，如果自动化测试在整个测试工作中发挥不了作用，那么哪怕是只投入了很少的成本也是在浪费资源。所以在开展自动化测试之前必须认真思考虑一些问题，以确保它的有效性：

1. 被测系统是否适合做自动化测试；

2、被测系统适合在哪些环节做自动化测试；

3、使用何种测试工具、测试框架；

4、开展自动化测试需要哪些资源，包括：人员、机器、时间；

5、当前可用或是可以申请到的资源；

6、如何在不影响日常测试工作的前提下，开展自动化测试工作；

****1、被测系统是否适合做自动化测试；****

很多人可能都回觉得这个问题有些多余，不需要考虑。可事实上作为一名测试主管必须考虑这个问题，应为自动化测试并非只是做与不做这么简单，而是需要考虑成本投入与效果产出。不管是做软件、做系统、还是做网站，最终的目的基本上都是要盈利的，要盈利就必须控制成本，提高利润。所以在测试工作中也必须考虑到成本投入。如何确定系统是否适合做自动化测试呢？个人认为主要考虑当前的被测项目是长期的还是短期？如果被测项目是在今后半年或者几年间要不断进行开发维护的，那么就需要重复的进行大量的回归测试，这种情况下如果有自动化的回归测试体系就可以节省成本投入；如果被测项目是短期的，比如说一个月，或几个月，同时如果开发自动化测试又不是很容易，也没有足够的时间去做这件事，那么就完全没有必要作自动化测试，费尽力气做出的自动化测试脚本，可能只会用到一次或两次，这样就很不值当。

****2、被测系统适合怎样的自动化测试；****

自动化测试覆盖的范围很广：单元测试、集成测试、接口测试，GUI测试等等都可以实现自动化执行；同时，不同的系统情况是不一样，有的适合或是可以做GUI的自动化测试,有的可能只适合做接口的自动化测试,所以需要针对不同的被测项目,考虑具体在哪一个环节作自动化测试。比如说针对搜索引擎，前端往往比较简单，只是一个文本框和提交按钮，大部分的逻辑处理都是在后端完成的，这种情况做自动化的接口测试就可以达到事半功倍的效果；如果是被测系统有很多的页面操作，那么可以考虑GUI的自动化测试；以上这两种情况都不是绝对的，如果测试资源足够，那么在各个环节都是可以开展自动化测试的。此外，还有一点需要考虑的是自动化测试的可行性，比如说对一个系统而言，做GUI测试是最合适的，也是最有效，但是有可能通过各种工具或者是脚本很难实现GUI的自动化测试，那么就需要考虑变通，考虑是否可以将自动化测试调整到接口测试或是集成测试等环节。

****3、使用何种测试工具、测试框架；****

当前可以做自动化测试的工具不少，主流的付费的qtp，winrunner，silktest等，开源免费的Jemmy,watir,pywinauto等等。这么多的工具框架该如何选择，就要看被测系统了，至少要考虑是C/S、还是B/S结构的，服务端和客户端都是用何种语言编写的，是什么数据库，服务器类型等等。

****4、开展自动化测试需要哪些资源，包括：人员、机器、时间；****

确定了使用何种测试工具、测试框架，就需要确定需要的资源，如：需要几个自动化测试工程师、需要购买的测试工具、测试机（服务器、客户机）、以及开发自动化框架所需要的时间；

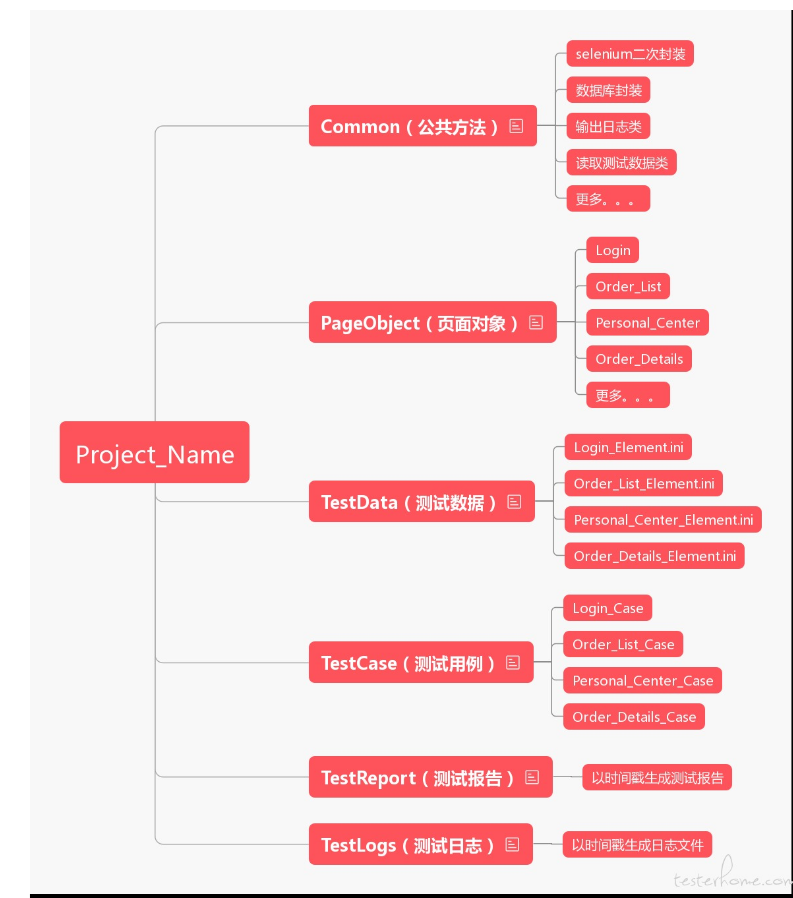
****5、当前可用或是可以申请到的资源；****

确定了需要哪些资源，就要看当前可用的资源有哪些，如果资源不足应提前申请，如招聘测试人员、购买配置测试机。甚至需要测试人员提前学习新的技术；

****6、如何在不影响日常测试工作的前提下，开展自动化测试工作；****

在以上问题考虑清楚以后，就需要制定自动化测试的计划进度了，这时候必须考虑到自动化测试的开展不能影响到日常的测试项目，毕竟不能因为开展自动化就把日常测试耽搁，该做的还是要做的，这点恐怕是必须要考虑的，整个公司或者整个项目往往不能为你特意分配一段时间做这样让某些人不能一下子看到和利润有何相关的事，所以必须安排好计划进度，更多的需要整个测试部门挤时间去做。

## 介绍一下你的自动化框架



## 8、各种设计用例，比如滴滴打车、微信聊天、二维码（突然发现桌子、笔、电梯、水杯啥的过时了......）

## 9、现场设计场景，说出查询服务端日志的命令？

**head命令**

1. Head -n -5 test.log:显示日志文件的开头5行；
2. Head -c 20 test.log:显示文件前20个字节；

**tail命令**

1. tail test.log：显示最后10行内容，tail默认显示最后10行内容；
2. tail -n 20 test.log：-n命令显示指定的行数；表示查看文件最后20行内容；
3. tail -f test.log：-f实时显示日志内容（默认10行，相当于添加参数-n 10）；
4. tail -n +10 test.log：从第10行开始显示内容；
5. tail -r -n 10 test.log：逆序显示最后10行内容；

**cat命令**

1. cat -n file：输出file文件中的全部内容,并且对file文件进行从 1 开始的编号；
2. cat -n text.txt  >  python.txt：把text.txt的文件内容加上行号后输入到python.txt中；
3. cat log.txt | grep ‘qq’:在log日志中查找qq字符；
4. cat log.txt | grep ‘qq’-A 5：在log日志中查找qq字符，并显示所在行之后5行；
5. cat log.txt | grep ‘qq’ -B 5：在log日志中查找qq字符，并显示所在行之前5行；
6. cat log.txt | grep ‘qq’ -C 5：在log日志中查找qq字符，并显示所在行前后5行；
7. cat log.txt | grep -v‘qq’：在log日志中查找排除qq字符所在的行；

**grep命令**

1. grep qq /home/test：在/home/test文件中查找单词qq；
2. grep qq /home/test /home/test1:在多个文件中查找单词qq；
3. grep -l qq /home/test /home/test1：使用-l参数，列出包含指定模式的文件的文件名；
4. grep -n qq /home/test：使用-n参数，在文件中查找指定模式并显示匹配行的行号；
5. grep -v qq /home/test：使用-v参数，输出不包含指定模式的行；输出/home/test文件中不包含单词qq的行；
6. grep ^qq /home/test：Bash[脚本](https://www.linuxcool.com/" \o "脚本" \t "https://www.linuxprobe.com/_blank)将 ^ 符号视作特殊字符，用于指定一行或者一个单词的开始。例如输出/home/test文件中所有以“qq”开头的行；
7. grep -i qq /home/test：使用-i参数，在查找时忽略字符的大小写；在/home/test文件中查找“qq”单词；
8. grep -e qq -e ww /home/test：使用-e参数，一条grep命令中在/home/test文件中查找‘qq和‘ww单词；
9. grep -B 4 qq /home/test：使用-B参数输出匹配行的前4行；
10. grep -A 4 qq /home/test:使用-A参数输出匹配行的后4行；
11. grep -C 4 qq /home/test：使用-C参数输出匹配行的前后各4行；
12. grep test \*file：查找后缀有 file 字样的文件中包含 test 字符串的文件，并打印出该字符串的行；
13. grep -r update /home：以递归的方式查找符合条件的文件。例如，查找指定目录/home及其子目录（如果存在子目录的话）下所有文件中包含字符串"update"的文件，并打印出该字符串所在行的内容；
14. grep -v test \*test\*：反向查找。前面各个例子是查找并打印出符合条件的行，通过"-v"参数可以打印出不符合条件行的内容。查找文件名中包含 test 的文件中不包含test 的行；

**more命令**

1. more +3 test.log:从第三行开始显示内容；
2. more -4 test.log:将日志内容设置为每屏显示4行；
3. more -5 +10 test.log：一屏5行，从第10行开始显示；
4. more -5 +qq test.log：先找到qq字符串，然后在这个字符串所在行的前两行开始显示；

**less命令**

1. less -n test.log：查看文件并显示行号；
2. less +10g test.log：定位到第10行；

**Vim命令**

**补充：**

跟tail功能类似的命令还有：

cat 从第一行开始显示档案内容。

tac 从最后一行开始显示档案内容。

more 分页显示档案内容。

less 与 more 类似，但支持向前翻页

head 只显示前面几行

tail 只显示后面几行

n 带行号显示档案内容

od 以二进制方式显示档案内容

## 输入网址后，发生了什么

**1.DNS解析**

DNS解析的过程就是寻找哪台机器上有你需要资源的过程，寻找的过程遵循就近原则。

输入一个网址并按回车的时候浏览器会根据输入的URL去查找对应的IP，具体过程如下：

（1）首先是查找****浏览器缓存****，浏览器会保存一段时间内访问过的一些网址的DNS信息，不同浏览器保存的时常不等。

（2）如果没有找到对应的记录，这个时候浏览器会尝试调用****操作系统缓存****来继续查找这个网址的对应DNS信息。

（3）如果还是没找到对应的IP，那么接着会发送一个请求到****路由器****上，然后路由器在自己的路由器缓存上查找记录，路由器一般也存有DNS信息。

（4）如果还是没有，这个请求就会被发送到ISP（注：Internet Service Provider，互联网服务提供商，就是网络运营商，中国电信中国移动等），ISP也会有相应的ISP DNS服务器，就是本地****DNS服务器****，请求的域名基本上都能在这里找得到。

（5）如果还是没有的话， ISP的DNS服务器会将请求发向****根域名服务器****进行搜索。根域名服务器就是面向全球的顶级DNS服务器，共有13台逻辑上的服务器，从A到M命名，真正的实体服务器则有几百台，分布于全球各大洲。

（6）如果到了这里还是找不到域名的对应信息，那只能说明一个问题：这个域名本来就不存在，它没有在网上正式注册过。或者域名过期了。

这也就是为什么有时候打开一个新页面会有点慢，因为如果本地没什么缓存，查找域名的过程要这样递归地查询下去，查找完还要一层层的向上返回。例如"mp3.baidu.com"，域名先是解析出这是个.com的域名，然后跑到管理.com域的服务器上进行进一步查询，然后是.baidu，最后是mp3， 所以域名结构为：三级域名.二级域名.一级域名。

所以DNS根据域名查询IP地址的过程为：浏览器缓存 --> 操作系统缓存 --> 路由器缓存-->本地（ISP）域名服务器缓存 --> 根域名服务器。

**2.进行TCP连接**

浏览器终于得到了IP以后，向服务器发送TCP连接，TCP连接经过三次握手。

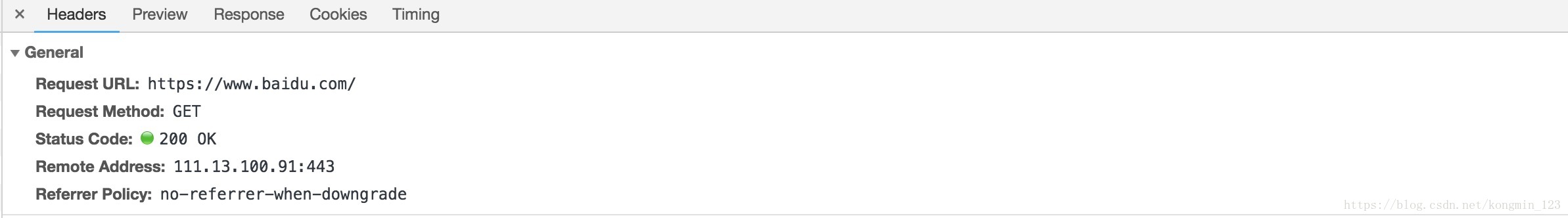
**3.浏览器发送HTTP请求**

浏览器和服务器建立连接以后，浏览器接着给这个IP地址给服务器发送一个http请求，方式为get，例如访问www.baidu.com。其本质是在建立起的TCP连接中，按照HTTP协议标准发送一个索要网页的请求。

这个get请求包含了主机（Host）、用户代理(User-Agent)，用户代理就是自己的浏览器，它是你的"代理人"，Connection（连接属性）中的keep-alive表示浏览器告诉对方服务器在传输完现在请求的内容后不要断开连接，不断开的话下次继续连接速度就很快了。可能还会有Cookies，Cookies保存了用户的登陆信息，一般保存的是用户的JSESSIONID，在每次向服务器发送请求的时候会重复发送给服务器。

在建立连接发送请求时每个服务端需要和客户端保持通信，有很多客户端都会和服务器进行通信。服务器为了识别是哪个客户端与它通信，就必须用一个标识记录客户端的信息。客户端首次访问服务器，服务端返回响应时通过附带一个记录的客户端信息的标识来返回给客户端，这个标识就是JSESSIONID，JSESSIONID就放在了客户端的Cookies里。当客户端再次向服务器发送请求时上就使用上次记录的Cookies里面的JSESSIONID，这样服务器就知道是哪个浏览器了。这样他们之间就能保持通信了。

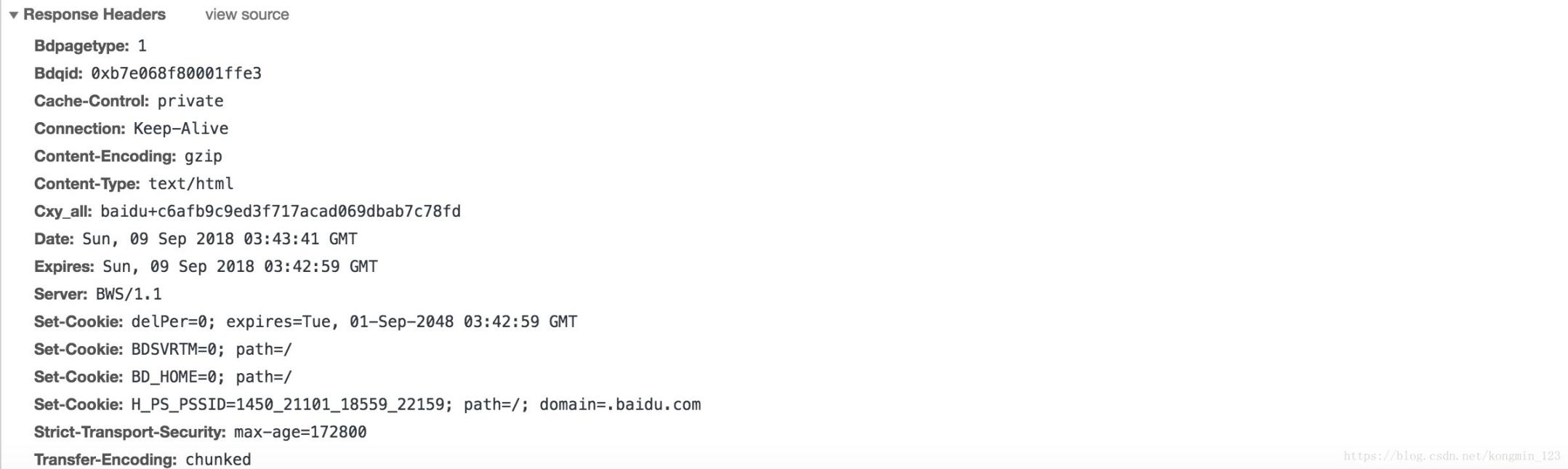
发送完请求接下来就是等待回应了，如下图：





**4.服务器处理请求**

发送完请求接下来就是等待回应了，如下图：



服务器收到浏览器的请求以后），会解析这个请求（读请求头），然后生成一个响应头和具体响应内容。接着服务器会传回来一个响应头和一个响应，响应头告诉了浏览器一些必要的信息，例如重要的Status Code，2开头如200表示一切正常，3开头表示重定向，4开头是客户端错误，如404表示请求的资源不存在，5开头表示服务器端错误。响应就是具体的要请求的页面内容。

**5.浏览器解析渲染页面**

（1）浏览器显示HTML

当服务器返回响应之后，浏览器读取关于这个响应的说明书（响应头），然后开始解析这个响应并在页面上显示出来。

浏览器打开一个网址的时候会慢慢加载这个页面，一部分一部分的显示，直到完全显示，知道最后的旋转进度条停止。因此在浏览器没有完整接受全部HTML文档时，它就已经开始显示这个页面了。

（2）浏览器向服务器发送请求获取嵌入在HTML中的对象

在浏览器显示HTML时，打开一个网页的过程中，主页（index）页面框架传送过来以后，浏览器还会因页面上的静态资源多次发起连接请求，需要获取嵌入在HTML中的其他地址的资源。这时，浏览器会发送一些请求来获取这些文件。这些内容也要一点点地请求过来，所以标签栏转啊转，内容刷啊刷，最后全部请求并加载好了就终于好了。

这时请求的内容是主页里面包含的一些资源，如图片，视频，css样式，JavaScript文件等等。

这在文件属于静态文件，首次访问会留在浏览器的缓存中，过期才会从服务器去取。缓存的内容通常不会保存很久，因为难保网站不会被改动。

静态的文件一般会从CDN中去取，CDN根据请求获取资源的时候可能还会用到负载均衡。

（3）浏览器发送异步（AJAX）请求

对于那些动态的请求，动态网页等就必须要从服务器获取了。对于静态的页面内容，浏览器通常会进行缓存，而对于动态的内容，浏览器通常不会进行缓存。对于这些动态请求，Nginx可能会专门设置一些服务器用来处理这些访问动态页面的请求。

**6.关闭TCP连接**

当数据完成请求到返回的过程之后，根据Connection的Keep-Alive属性可以选择是否断开TCP连接，HTTP/1.1一般支持同一个TCP多个请求，而不是1.0版本下的完成一次请求就发生断开。TCP的断开与连接不一样，断开可以分为主动关闭和被动关闭，需要经过4次握手。

当浏览器需要的全部数据都已经加载完毕，一个页面就显示完了。

## 11、所有问到我 mock  的面试官都不知道 Postman  搭建  mockserver

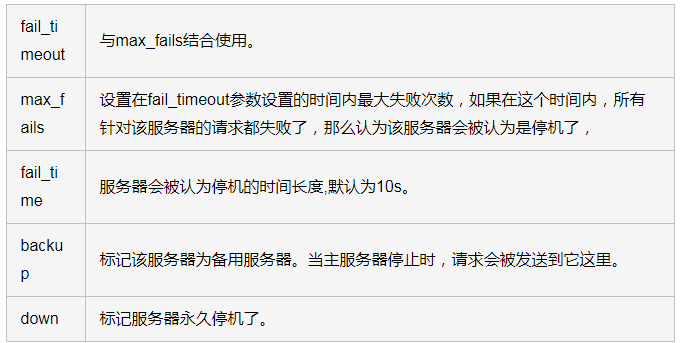
## 12、Nginx负载均衡策略



**1、轮询**

　　最基本的配置方法，上面的例子就是轮询的方式，它是upstream模块默认的负载均衡默认策略。每个请求会按时间顺序逐一分配到不同的后端服务器。

　　有如下参数：



注意：

* 在轮询中，如果服务器down掉了，会自动剔除该服务器。
* 缺省配置就是轮询策略。
* 此策略适合服务器配置相当，无状态且短平快的服务使用。

**2、weight**

　　权重方式，在轮询策略的基础上指定轮询的几率。例子如下：



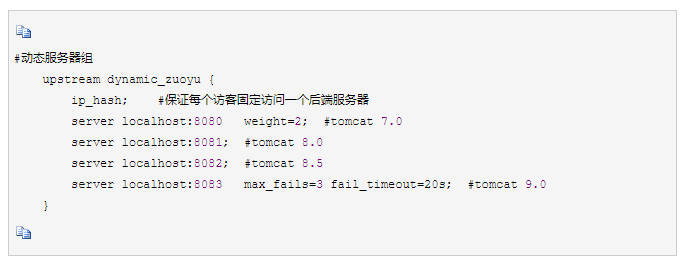
　　在该例子中，weight参数用于指定轮询几率，weight的默认值为1,；weight的数值与访问比率成正比，比如Tomcat 7.0被访问的几率为其他服务器的两倍。

　　注意：

* 权重越高分配到需要处理的请求越多。
* 此策略可以与least\_conn和ip\_hash结合使用。
* 此策略比较适合服务器的硬件配置差别比较大的情况。

**3、ip\_hash**

　　指定负载均衡器按照基于客户端IP的分配方式，这个方法确保了相同的客户端的请求一直发送到相同的服务器，以保证session会话。这样每个访客都固定访问一个后端服务器，可以解决session不能跨服务器的问题。



　　注意：

* 在nginx版本1.3.1之前，不能在ip\_hash中使用权重（weight）。
* ip\_hash不能与backup同时使用。
* 此策略适合有状态服务，比如session。
* 当有服务器需要剔除，必须手动down掉。

**4、least\_conn**

　　把请求转发给连接数较少的后端服务器。轮询算法是把请求平均的转发给各个后端，使它们的负载大致相同；但是，有些请求占用的时间很长，会导致其所在的后端负载较高。这种情况下，least\_conn这种方式就可以达到更好的负载均衡效果。



　　注意：

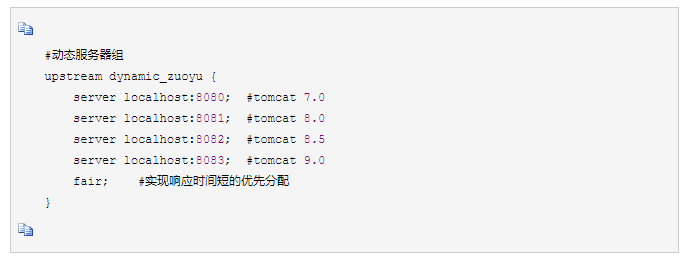
* 此负载均衡策略适合请求处理时间长短不一造成服务器过载的情况。

**5、第三方策略**

　　第三方的负载均衡策略的实现需要安装第三方插件。

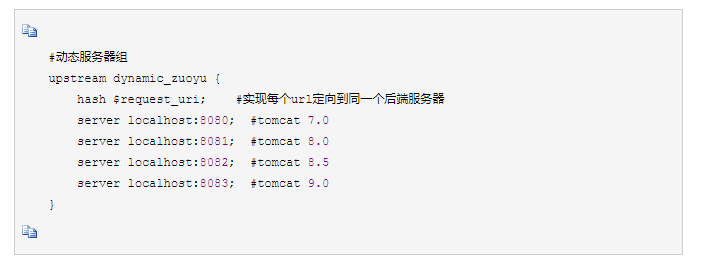
**①fair**

　　按照服务器端的响应时间来分配请求，响应时间短的优先分配。



**②url\_hash**

　　按访问url的hash结果来分配请求，使每个url定向到同一个后端服务器，要配合缓存命中来使用。同一个资源多次请求，可能会到达不同的服务器上，导致不必要的多次下载，缓存命中率不高，以及一些资源时间的浪费。而使用url\_hash，可以使得同一个url（也就是同一个资源请求）会到达同一台服务器，一旦缓存住了资源，再此收到请求，就可以从缓存中读取。



## 13、为什么要写用例

1. 理清思路，避免漏测和重复测；
2. 提高测试效率；
3. 跟进测试进度；
4. 更好的发现问题，记录问题，复现问题；
5. 跟进重复性工作；
6. 告诉领导：我做过；
7. 接口测试流程中的一个产物（测试用例）

## 14、怎么进行接口测试

通过工具模拟客户端向服务端发送请求并接受服务器返回的数据来对接口的功能，逻辑业务，异常，安全进行测试；

（1）功能测试：测试这个接口的功能是否实现，并且测试这个接口是否按照接口文档来进行开发的（比如说接口文档规定了一些关键字，而开大的时候把关键字改成了其他的关键字，因为在整个项目周期，并不只有一个开发而是有多个，所以可能因为在开发过程中因为关键字不一样导致某些开发的功能异常，还有自动化脚本也会发生异常）

（2）逻辑业务，主要指的是一些逻辑业务依赖关系（比如支付宝提交订单的时候要保证你是在登录的情况下，如果你没有登录而提交成功了，这就是异常，可以修改请求的cookie来测试）

（3）异常测试：参数异常：关键字参数（应用其他的关键字替换进行测试）、参数为空、参数多少（通过添加参数增添个数），参数错误。数据异常：关键字数据（填入的数据用其他的数据语言的数据替用）、数据长度、数据为空、数据错误。  
　　  
　　由于我们项目前后端调用主要是基于http协议的接口，所以测试接口时主要是通过工具或代码模拟http请求的发送与接收。工具有很多如：postman、jmeter、soupUI、java+httpclient、robotframework+httplibrary等。  
　　–也可以用 接口自动化来实现，就是用代码实现，框架和UI自动化差不多，发送请求用断言来判断。

## 15、没有接口文档，如果做接口测试？（这是个送命题）

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

## 16、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

## 17、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

## 18、当一个接口出现异常时候，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

## 19、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

## 20、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

## 21、在实际项目中你是如何做测试计划

1.对客户提供的或需求分析人员编写的用户需求文档或需求规格说明书进行分析，提炼出测试要点；

2.根据测试要点编写测试用例。

3.由评审组对测试用例进行评审--修改--再次评审--初步定稿

4.执行测试

4.1 按照测试用例对系统进行功能验证及客户的需求验证

4.2 将测试过程中产生的Bug录入缺陷管理系统

4.3 新版本发布后，对本次版本新增加的功能以及开发人员修正的Bug进行回归测试

4.4 根据项目需要提交测试报告。

# 二、面试问题：

## 2.1

### 1，测试这个系统对于我们测试来说，你觉得最大的挑战是什么？最复杂的模块？

熟悉业务，只有熟悉业务，才能充分全面的设计测试用例，覆盖所有的功能，最后才能保证项目的质量；

### 2，第三方接口依赖，用 Python 写的 mock 服务吗？主要写的哪几个？ 做什么用的啊？



### 3，postman 怎么搭建 mock 服务？

1. 打开postman,新建一个mocke服务；
2. 自定义方法的请求方法（post get delete等），接口名称，请求包体，接口描述、接口返回代码,接口返回数据；
3. mockserver会生成一个域名，向这个域名+自定义的接口进行发送请求，返回结果代码以及结果内容就是之前定义好的内容；

### 4，挑一个你项目中你负责的比较小的模块并且跟我介绍一下，然后说一下你针对这个模块的测试设计了哪些异常用例？

学生信息绑定模块—学籍号正常16位

异常用例：

1. 学籍号为空，相应提示信息；
2. 学籍号少于16位，相应提示信息；
3. 学籍号输入字符、特殊字符，相应提示信息；
4. 未存在的学籍号，相应提示信息；
5. 重复绑定学籍号，数据是否重复；
6. 不同网络条件下学籍号绑定；
7. 大数据量下绑定的响应时间；

### 5，微信发红包的测试用例？

****功能****  
1.在红包钱数，和红包个数的输入框中只能输入数字  
2.红包里最多和最少可以输入的钱数  200  0.01  
3.拼手气红包最多可以发多少个红包  100  
3.超过最大拼手气红包的个数是否有提醒  
4.当红包钱数超过最大范围是不是有对应的提示  
5.当发送的红包个数超过最大范围是不是有提示  
6.当余额不足时，红包发送失败  
7.在红包描述里是否可以输入汉字，英文，符号，表情，纯数字，汉字英语符号，  
7.1是否可以输入它们的混合搭配  
8.输入红包钱数是不是只能输入数字  
9.红包描述里许多能有多少个字符   10个  
10.红包描述，金额，红包个数框里是否支持复制粘贴操作  
12.红包描述里的表情可以删除  
13.发送的红包别人是否可以领取  
13.发的红包自己可不可以领取   2人  
14.24小时内没有领取的红包是否可以退回到原来的账户  
14.超过24小时没有领取的红包，是否还可以领取  
15.用户是否可以多次抢一个红包  
16.发红包的人是否还可以抢红包   多人  
17.红包的金额里的小数位数是否有限制  
18.可以按返回键，取消发红包  
19.断网时，无法抢红包  
20.可不可以自己选择支付方式  
21.余额不足时，会不会自动匹配支付方式  
22.在发红包界面能否看到以前的收发红包的记录  
23.红包记录里的信息与实际收发红包记录是否匹配  
24.支付时可以密码支付也可以指纹支付  
25.如果直接输入小数点，那么小数点之前应该有个0  
26.支付成功后，退回聊天界面  
27.发红包金额和收到的红包金额应该匹配  
28.是否可以连续多次发红包  
29.输入钱数为0，"塞钱进红包"置灰

****性能****  
1.弱网时抢红包，发红包时间  
2.不同网速时抢红包，发红包的时间  
3.发红包和收红包成功后的跳转时间  
4.收发红包的耗电量  
5.退款到账的时间

****兼容****  
1.苹果，安卓是否都可以发送红包  
2.电脑端可以抢微信红包

****界面****  
1.发红包界面没有错别字  
2.抢完红包界面没有错别字  
3.发红包和收红包界面排版合理，  
4.发红包和收到红包界面颜色搭配合理

****安全****  
1.对方微信号异地登录，是否会有提醒   2人  
2.红包被领取以后，发送红包人的金额会减少，收红包金额会增加  
3.发送红包失败，余额和银行卡里的钱数不会少  
4.红包发送成功，是否会收到微信支付的通知

****易用性(有点重复)****  
1.红包描述，可以通过语音输入  
2.可以指纹支付也可以密码支付

## 2.2

### 第一轮  技术官

### 1，自我介绍

### 2，简单介绍最近的工作以及项目

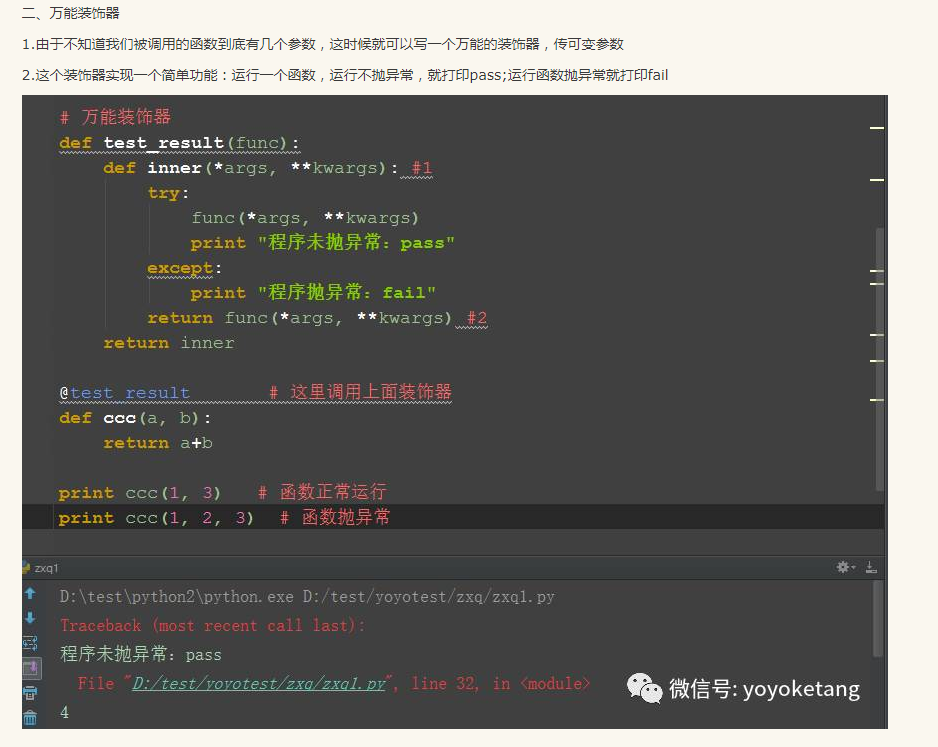
### 3，装饰器有了解吗？简单介绍一下装饰器

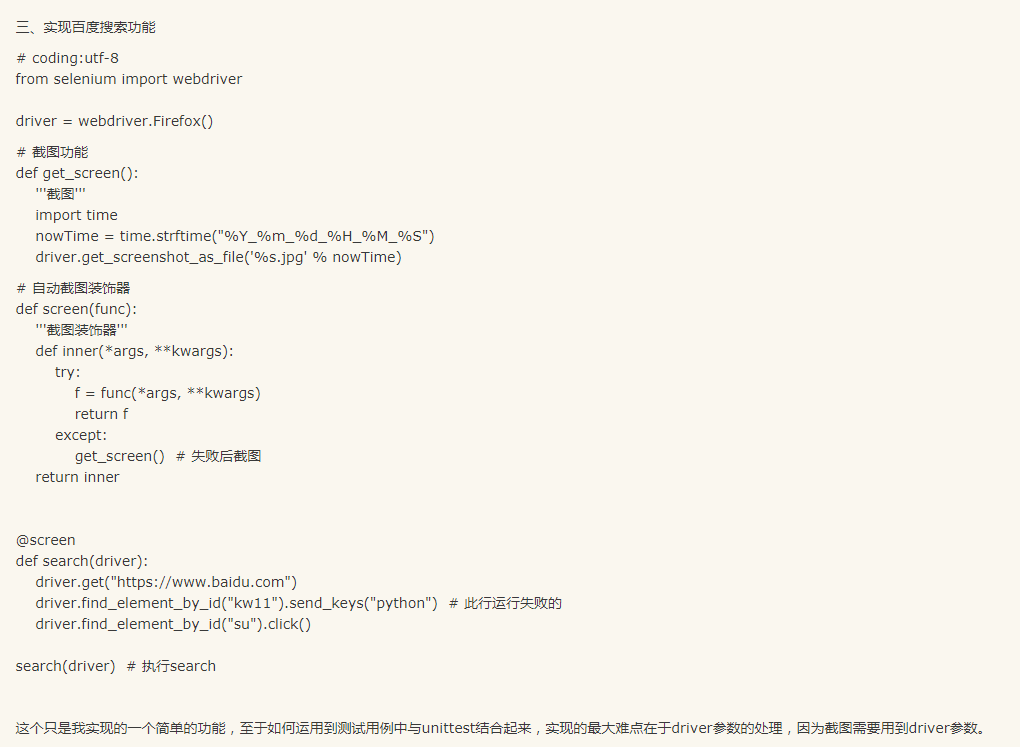
装饰器是将另一个函数作为输入并修改/扩展其功能的函数；可以抽离出大量与函数功能本身无关的雷同代码并继续重用；

### 有用过什么样的装饰器？举个例子

1、[Python3+selenium-unittest之装饰器（@classmethod）](https://www.cnblogs.com/EncoreLiu/p/10742565.html)

setup():每个测试case运行前运行  
teardown():每个测试case运行完后执行  
setUpClass():必须使用@classmethod 装饰器,所有case运行前只运行一次  
tearDownClass():必须使用@classmethod装饰器,所有case运行完后只运行一次





### 5，测试登录，然后发送一条弹幕，这样的一个场景你会怎么设计自动化测试用例？

### 6，(回答 context 上下文获取 ID token)用前置后置会怎么操作？

### 7，有了解过公钥私钥吗？

一、****公钥****加密   
假设一下，我找了两个数字，一个是1，一个是2。我喜欢2这个数字，就保留起来，不告诉你们(私钥），然后我告诉大家，1是我的公钥。

我有一个文件，不能让别人看，我就用1加密了。别人找到了这个文件，但是他不知道2就是解密的私钥啊，所以他解不开，只有我可以用  
数字2，就是我的私钥，来解密。这样我就可以保护数据了。

我的好朋友x用我的公钥1加密了字符a，加密后成了b，放在网上。别人偷到了这个文件，但是别人解不开，因为别人不知道2就是我的私钥，  
只有我才能解密，解密后就得到a。这样，我们就可以传送加密的数据了。

二、****私钥****签名  
如果我用私钥加密一段数据（当然只有我可以用私钥加密，因为只有我知道2是我的私钥），结果所有的人都看到我的内容了，因为他们都知  
道我的公钥是1，那么这种加密有什么用处呢？

但是我的好朋友x说有人冒充我给他发信。怎么办呢？我把我要发的信，内容是c，用我的私钥2，加密，加密后的内容是d，发给x，再告诉他  
解密看是不是c。他用我的公钥1解密，发现果然是c。  
这个时候，他会想到，能够用我的公钥解密的数据，必然是用我的私钥加的密。只有我知道我得私钥，因此他就可以确认确实是我发的东西。  
这样我们就能确认发送方身份了。这个过程叫做数字签名。当然具体的过程要稍微复杂一些。用私钥来加密数据，用途就是****数字签名****。

三、公钥，就是给大家用的，你可以通过电子邮件发布，可以通过网站让别人下载，公钥其实是用来加密/验章用的。私钥，就是自己的，必须非常小心保存，最好加上密码，私钥是用来解密/签章，首先就Key的所有权来说，私钥只有个人拥有。公钥与私钥的作用是：用公钥加密的内容只能用私钥解密，用私钥加密的内容只能用公钥解密。

比如说，我要给你发送一个加密的邮件。首先，我必须拥有你的公钥，你也必须拥有我的公钥。

首先，我用你的公钥给这个邮件加密，这样就保证这个邮件不被别人看到，而且保证这个邮件在传送过程中没有被修改。你收到邮件后，用你的私钥就可以解密，就能看到内容。

其次我用我的私钥给这个邮件加密，发送到你手里后，你可以用我的公钥解密。因为私钥只有我手里有，这样就保证了这个邮件是我发送的。

### 8，pytest 有用过吗？

### 9，list 怎么去重？

1、使用两个for循环遍历集合所有元素，然后进行判断是否有相同元素。两两比较如果相等的话就删除第二个。

2、背过特性的都知道HashSet可以去重，把List集合所有元素存入HashSet对象，接着把List集合元素全部清空，最后把HashSet对象元素全部添加至List集合中，这样就可以保证不出现重复元素。

3、对于大数据，采用Stream相关函数去重是最简单的了。Java8中Stream提供了对List 做简单去重的处理，通过调用distinct方法，可以实现对类型Integer、Long、Char 等基本类型以及String类型的去重。需要注意的是，无法对自定义对象进行去重处理。

4、Lambda表达式是Java8中提供的一种新的特性，它支持JAVA也能进行简单的“函数式编程”。利用Lambda表达式能很方便的对list对象进行去重，而且不会造成代码入侵。不过，Lambda表达式只能出现在目标类型为函数式接口的上下文中。

### 有一个字典，反转 key-value

a={1:10,2:20,3:30}

b={v:k for k,v in a.items()}

print()

print(a.items())

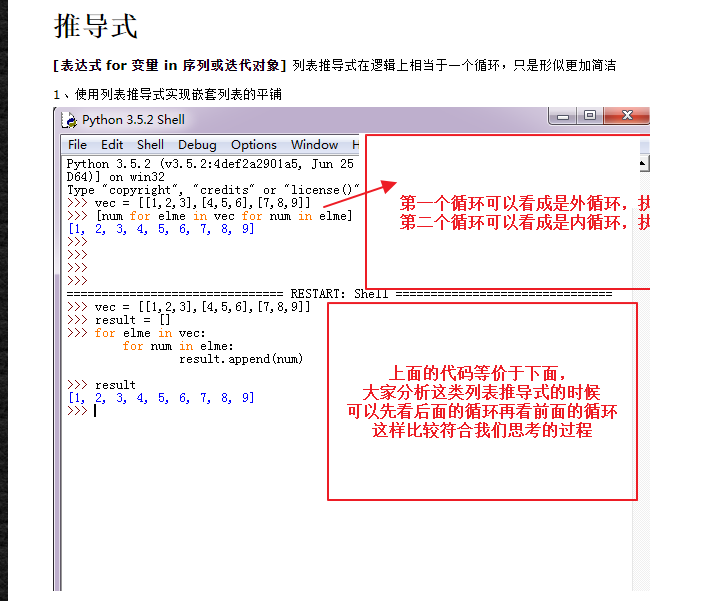
print(b)

### 11，实际工作中有对哪些数据做过格式化处理？用过哪些数据类型？

### 12，有用过列表推导式吗？

回来查下推导式，原来就是常用的list的for in for，有点想笑，这么简单的列表创建问题问得让我摸不着头脑，有些术语听起来很深奥，其实就是常用的表达式，这好比你去问别人设计模式，你用过什么设计模式，什么工是工厂模式？什么是单例模式等等，虽然对术语可能一知半解或者解释不清，其实就是常常在用的。

newlist = [i for i in range(1, 100) if i%3==0]

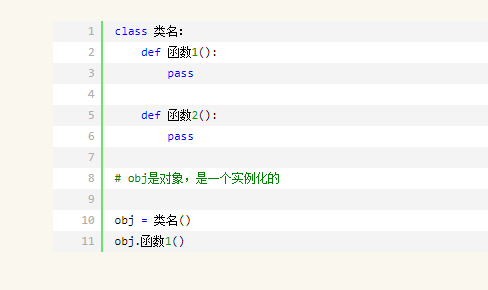


### 13，有了解过生成器 generator 之类的吗？

### 14，你对 python 面向对象有什么概念？能举一个测试工作中用到面向对象的场景吗？

Selenium框架中提取公共功能，比如登录功能；说明：我们把一些公共的功能，可以提取出来，并且在公共的功能中创建属于这个对象的属性，然后其他的方法就可以使用这个对象的属性了

面向对象编程其实就是：类 + 对象





### 你测试的话，会怎么样设计测试用例？就是说测试用例会有哪些文件？比如说正常的 python 文件和正常的数据文件，还有其他的吗？

测试报告文件、测试日志文件、

### 有了解 unittest 执行逻辑，就是整个生命周期吗？



生命周期：  
setupclass>setup>test01\_login>test\_teardown 这是第一个循环；  
setupclass>setup>test02\_demo>test\_teardown>teardownclass 第二次循环；

setUp()中建立数据库连接以及进行一些初始化，在tearDown()中清除在数据库中产生的数据，然后关闭连接。

### 有用过 Git 吗？







### 数据驱动你是怎么理解的？

使用数据驱动的模式，可以根据业务分解测试数据，只需定义变量，使用外部或者自定义的数据使其参数化，从而避免了使用之前测试脚本中固定的数据。可以将测试脚本与测试数据分离，使得测试脚本在不同数据集合下高度复用。不仅可以增加复杂条件场景的测试覆盖，还可以极大减少测试脚本的编写与维护工作。

Python下的数据驱动模式(ddt)库，结合unittest库以数据驱动模式创建测试用例。

ddt库包含一组类和方法用于实现数据驱动测试。可以将测试中的变量进行参数化。

### 你是用什么实现的引入外部数据集合？

通过Python下的数据驱动模式(ddt)库，获取外部excel数据，对变量进行参数化；

### 假如有这样一个场景，一个登陆我们写 10 条 case，但是逻辑都是一样的，但是自动化的话我不可能同样的逻辑赋值十遍，那你有什么办法来解决这个问题？

业务逻辑是相同的，其中登录的测试数据是不同的；可以使用数据驱动模式，比如python下ddt库对测试数据进行变量参数化；

### 第二轮  用人部门技术官

### 1，先简单介绍一下自己的情况

### 2，你当时用的是 python 的什么测试框架还是？

Selenium+python+unittest框架；

### 3，所有的调用流程都是你自己写的吗？

### 4，unittest 在运行过程中有一些不优美的地方，你当时在项目中使用的时候有没有发现什么？

执行顺序是根据ASCII码的顺序加载测试用例，数字与字母的顺序为：0-9，A-Z，a-z；只能通过TestSuite类的addTest方法，按顺序加载测试用例；或者通过修改函数名的方式；

### 5，unittest 和 pytest 各有什么优缺点？





### 你当时用的是 unittest 框架，那你的用例和数据是怎么去隔离开的？

DDT数据驱动模式；业务逻辑和测试数据分离；通过ddt类库读取excel中的数据实现变量的参数化，达到不同测试用例的运行；

### HTMLTestRunner 这个第三方库的报告还是比较简陋的，你当时生成的报告里面主要体现了哪些东西？

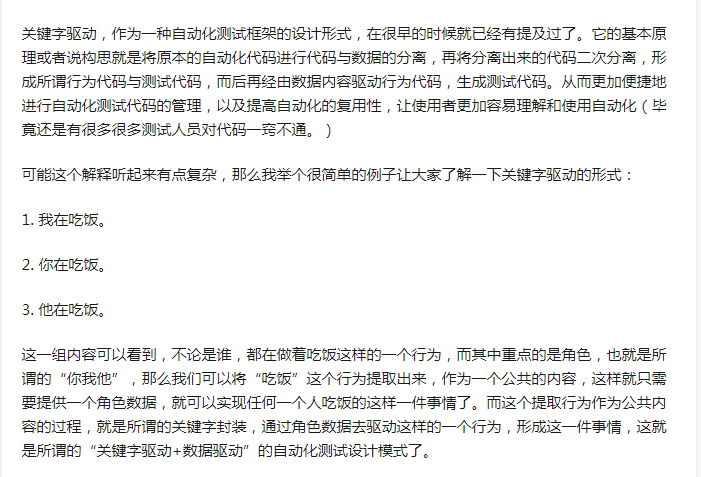
测试人员；开始时间；合计耗时；测试结果统计（通过，失败，通过率）；每条失败用例的断言信息；

### 8，你在项目中设计了有多少条用例？用例级别？

### 9，HTML 测试报告怎么可以看起来更直观？一个不懂你项目的人看到你的测试报告也能立马知道你出错的原因是什么？？

用Pytest+Allure生成可视化的HTML测试报告；框架代码中设计try exception和断言，报告中显示失败的结果信息；

### 有用到关键字驱动吗？



### 11，你对 Python的掌握怎么样？

### 12，Python实际应用十分制的话，你给自己打多少分？

### 13，两个接口的返回值要做一个字段的对比，你用 Python 会怎么去做？



### 字符串排序你要怎么做？



### 你做接口自动化和手工测试，大概占比是多少？

### 你们当时项目自动化只对接口的返回值返回内容做校验吗？

还有数据库校验，比如新增数据接口，我接口返回了成功，但是是否真的新增成功还是要检验数据库里面的数据；当然，这样就跟功能测试一样了。我认为接口测试对这个问题的验证可以通过别的接口完成，比如有删除的api就有状态检查的api，你删除前先看下这个接口的返回值，删除之后再查一遍，看看少的是不是你删掉的哪个；

# 三、面试心得：

首先最重要的，不要对自己所学的知识产生怀疑，有句话说得好，要站在巨人的 肩膀上。你会使用那些功能强大的库已经很不错了所以信心很重要！当你怀疑自己能力时，无意中就会使自己在沟通的过程中变得畏畏缩缩，深陷此坑，持续数场面试，直到遇到这位面试官一段话把我点醒。我是来应聘工作的，而非来做学术研讨， 懂的多那是加分项，即使不懂那也是理所应当，心态放稳。