# 测试流程及规范

## 如何提高测试的质量：

1.需求与测试需求方面  
　　1.应从产品人员手中获取需求，尽量要求产品人员对需求进行讲解（需求评审）  
　　2.我们要做测试需求分析，并应用相对应的方法论，使用xmind分解功能点，尽可能多的列出其有效场景和无效场景，还有特殊场景。

3.我们把输入项有效取值的场景叫做有效场景，无效取值的场景叫做无效场景，从其他角度进行需求分析的场景叫做特殊场景。

4.常见的特殊场景有：重复操作（例如重复进行无效操作），权限不足，流程回退操作，**业务分析**。  
2.测试计划  
　　1.计划影响的是执行，计划需要根据实际情况来做，计划一旦制定就需要尽量的按照这个计划执行。

2.测试计划的三种实现方式：传统计划（时间人员任务安排的进度计划），任务跟踪表（记录三个关键节点），禅道看板。【注：任说其中一种情况即可】  
3.用例设计  
　　1.所有功能点和需求规则都要被覆盖，（用例设计方法、评审、需求分析）  
　　2.最少的测试用例，（合适的用例设计方法）  
　　3.应多考虑业务层面的测试用例，安全、性能等各种非功能层面也应该考虑。（全栈）

4.但是用例会存在一部分功能不写用例，类似于界面类、按钮类，功能重叠类，兼容性用例，不写用例，但需要进行测试。

5.用例写作时，一句话的用例，用xmind写用例值得推广，因为可以提高效率。  
　　6.多考虑测试场景（有效场景、无效场景、特殊场景）与结果检查  
　　7.应对测试用例进行评审（评审是判断测试人员的测试范围和测试策略的最佳方法）

4.测试准备  
　　1.测试环境应该要尽可能的接近真实环境  
　　2.测试环境应该能够包含各种测试场景  
　　3.测试数据应该尽可能真实，也包含了各种能够覆盖场景的测试数据  
5.测试执行  
　　1.用例要100%的执行，不能够放过任何一个测试用例或测试场景  
　　2.测试的轮次安排，应安排2轮测试。总测试周期约为2周，测试执行周期约为1周。

3.每一轮测试重点应该不一样，第一轮测试的重点是界面+功能，第二轮测试会更换浏览器（手机），并且更加关注业务功能测试甚至结合Fiddler、数据库等工具进行辅助，界面在本轮不再是测试的绝对重点。  
　　4.使用测试工具（数据库工具，Linux、自动化、接口、性能测试等），  
　　5.引用缺陷管理工具（禅道或JIRA），标准化缺陷管理流程，规范开发与测试对缺陷的管理  
　　6.过Bug是个好东西  
　　7.让产品人员、客户也介入测试，进行UAT测试  
　　8.让开发人员也介入测试，让他们做单元测试（开发内测）

6.测试总结  
　　似乎与质量没有直接关系  
7.测试管理相关  
　　1.编写测试日报，定期向团队汇报测试进展以及问题及风险  
　　2.为了让文档的管理变得更规范，应引入SVN进行配置管理

## 详细的描述一个测试活动完整的过程。

 1. 项目经理通过和客户的交流，完成需求文档，由开发人员和测试人员共同完成需求文档的评审，评审的内容包括：需求描述不清楚的地方和可能有明显冲突或者无法实现的功能的地方。项目经理通过综合开发人员，测试人员以及客户的意见，完成项目计划。然后SQA进入项目，开始进行统计和跟踪

 2. 开发人员根据需求文档完成需求分析文档，测试人员进行评审，评审的主要内容包括是否有遗漏或者双方理解不同的地方。测试人员完成测试计划文档，测试计划包括的内容上面有描述。

3. 测试人员根据修改好的需求分析文档开始写测试用例，同时开发人员完成概要设计文档，详细设计文档。此两份文档成为测试人员撰写测试用例的补充材料。

4.测试用例完成后，测试和开发需要进行评审。

 5. 测试人员搭建环境

 6. 开发人员提交第一个版本，可能存在未完成功能，需要说明。测试人员进行测试，发现BUG后提交给禅道管理系统。

 7. 开发提交第二个版本，包括Bug Fix以及增加了部分功能，测试人员进行测试。

 8. 重复上面的工作，一般是3-4个版本后BUG数量减少，达到出货的要求。

 9. 如果有客户反馈的问题，需要测试人员协助重现以及回归测试。

## 怎样做好测试计划？

1)理解系统。从整个系统的高度了解被测系统必须满足的功能和非功能性需求。利用涉及整个系统的文档，形成对系统的整体了解。

2)及早介入。为了深入了解项目，测试人员应该在系统的开始阶段介入，可以增加对客户需求，客户问题，潜在风险，以及最重要的功能方面的理解

3)测试期望。程序员的期望是什么？客户的期望是什么？销售对测试的期望又是什么？测试目标必须是绝对的，以免说不清楚是否达到目标。

4)吸取教训。把以前工作中学习到的经验教训运用过来，对确定测试策略很有作用。

5)工作量大小。完成测试需要多少工作量？需要多少人员？

6)技术选择。系统会采取什么技术？系统会采用什么架构？这些信息有助于确定测试策略和测试工具。

7)时间表。系统开发和测试分配的时间有多长？截止日期是什么时候？

## 测试日报的内容

测试日报一般在测试执行过程中，由测试人员编写汇报给项目团队测试的进展情况。

测试日报的内容主要有：

1. 当天的工作总结（执行了哪些模块、执行了多少用例、发现了多少缺陷等）
   1. 建议站在项目的角度进行说明，说明应该包含总体进度。
   2. 一般工作进度有乐观和悲观的情况，一般要表现出情况是乐观的还是悲观的。
   3. 如果情况是悲观的，则应该说明因为一些什么原因导致了结果的悲观。
2. 次日的工作计划
3. 问题和风险说明
   1. 如果在工作中遇到了一些问题会阻碍工作的开展，请描述出来并且写出你所期望的解决方案。

## 小谈UAT（验收测试）

验收测试人员的测试任务：

1.验收人员是提出需求的人员，所以对需求最为熟悉，最主要测试功能的遗漏或者多余  
2.系统测试人员重点在测试功能的正确性和非功能的符合性，当然也希望验收人员测试功能的正确性  
3.因为验收人员也是往后系统的使用人员，请验收人员务必关注用户体验和易用性

系统测试人员给验收测试人员提供的帮助：

1.不需要给验收测试人员提供理解需求的工作  
2.验收测试人员测试环境和测试数据  
3.在验收人员发现缺陷后，可以提供相应的协助。（联系开发，修复缺陷）

## 测试总结的写作要点

测试总结就是测试报告，测试报告是有模板的。模板中包含的内容：

1.测试的数据说明，你测试了哪些功能点，执行了哪些用例，发现多少缺陷，采用了哪些方法。

2.对软件质量的评估，评估是好还是不好。

3.测试的结论，通过还是不通过？如果有遗留问题或者其他问题，需要进行评审是否会给用户造成较大风险。

## 测试结束标准

1）一二级缺陷数目达到项目质量管理目标要求，测试暂停返回开发

2）项目出现重大估算和进度偏差，需要暂停或者终止

3）新需求变更大，需修改测试计划和测试用例再进行

4）开发暂停，测试也暂停，备份暂停时的数据

5）所有功能、性能测试用例100%进行

## 功能测试/测试流程（规范）：

1.过bug（亮点）：

1.防止开发人员不理会Bug，防止拖延

2.防止没人负责，不愿意负责

2.写测试日报（亮点）

1.能够告诉领导，你做了哪些工作，计划做哪些工作。领导放心

2.对于编写人员好处是：知会问题，请求帮助。

3.用例评审（亮点）

1.用例的价值：是研究这个功能/系统要测试哪些内容，如何测试。

2.与相关成员（产品、开发）沟通是否测试这些内容，是否测对了。

4.需求串讲（执行有难度）

1.产品人员讲解需求之后，由测试人员再讲一遍，根据测试思维再讲一遍。

5.编写用例

1.规范的用例是比较难写的，也是测试团队看重的。

2.用例的价值：不是所有的都应该写用例。

3.如何做到质量和数量的平衡（亮点）

6.用例设计方法

1.超喜欢说的等价类、边界值、判定表、场景法-->组合

2.有效场景、无效场景、特殊场景（亮点）

7.测试计划与方案

1.方案是解决怎么测的问题，然而在用例中也是解决这个问题。所以方案没什么用

2.计划的内容：时间人员任务安排（核心），准入准出标准（都一样），风险相关（当场就报）。

8.任务跟踪表（亮点）

1.以任务形式来编写计划和相关内容

2.任务跟踪表的内容：

1.每个大阶段的计划时间、实际开始时间

2.相关干系人说明

3.任务状态描述

9.测试执行

1.如果只是功能测试，手工点点点，绝对Low。

2.就算是点点点，也有策略。第一轮测什么，第二轮测什么，回归怎么搞，冒烟怎么做。（亮点）

3.工作不仅仅是手工的，还要有工具支持（想办法代入）：

1.数据库SQL：

1.结果的查看，要用查询语句

2.制造特殊数据，修改数据，用到增、删、改。-->提高效率

3.生产问题跟踪，用到查询语句。

2.用到抓包，Fiddler

1.定位问题，去抓响应数据，查看响应数据。

2.发现问题，做简单的接口测试

3.自动化、接口测试、性能测试的代入

1.接口测试过程、工具的使用、接口测试哪些模块

2.性能测试线程数、JMeter使用、结果（响应时间、事务成功率、nmon）

4.业务的闭环（亮点）

1.闭环思维的关词：偏差和反馈，发现偏差并及时反馈，这正是项目经理 的工作重点，也是项目管理中最难的环节.

5.跟踪问题（不好理解）

1.日志相关：Linux服务器日志和Android日志（adb logcat）

## BS架构和CS架构的区别

BS架构即为：Browser（浏览器）Server（服务器）架构，也就是在浏览器中打开的应用系统，即WEB系统。CS架构为：Client（客户端）Server（服务器）架构，在终端需要安装一个应用程序方可访问的应用系统。一般在手机上极为常见，App就是一种特殊的CS架构。

**B/S与C/S的区别：**

1. 安装：B/S不需要安装，直接通过浏览器打开，C/S需要安装客户端。
2. C/S客户端因为是独立设计，所以可实现个性化。B/S不可以，但能通过账号体系进行个性化设置。
3. 安全性：C/S采用私有协议并且客户端独立设计可以引入安全性策略，B/S采用公共协议（HTTP协议），理论上C/S更安全。实际上B/S系统其实也足够安全，因为可以使用安全的HTTPS协议，安装控件的方式实现较好的安全性。
4. C/S需要独立开发客户端，其开发成本更高。
5. C/S运行从本地加载客户端，只需要加载少量网络数量，更节约流量。这是APP采用C/S架构的原因之一
6. 维护性：B/S不需要用户手动维护升级，C/S需要定期维护升级（APP经常1个月就需要升级一次）

兼容性：C/S考虑浏览器的兼容性，C/S考虑操作系统的兼容性

## 如何进行自我介绍

自我介绍在面试中常常作为第一个问题而出现，好的自我介绍可以带来良好的第一印象，如何讲好自我介绍呢。我建议从三部分下手

1.我是谁

2.我会什么

3.我想做什么

----------------------------范例----------------------------

【我是谁】面试官您好，我叫XXX，来自湖南。2013年毕业于XXX大学，有2年软件测试工作经验，之前在XXX公司担任软件测试工程师一职。

【我会什么】在公司里我先后负责了两个项目的测试，分别为XX银行现金管理项目和移动BOSS开发维护项目，在这两个项目中我负责了测试计划和方案的编写，测试用例的设计，测试环境的搭建以及测试执行和编写测试报告等工作。在工作中我组要负责功能测试，其次还参与了一些非功能测试，如兼容性测试，易用性测试，性能测试等。

【我想做什么】今次我来贵公司是求职软件测试工程师，希望得到这样一个机会。

----------------------------------------------

注：在【我会什么】部分中如果觉得内容很简短，可以对相关知识进行拓展来说。 在【我想做什么】部分中可以弱化，因为面试的目的本身就是求职，不需要特别强调。

建议：在自我介绍中多说自己会什么，自己做过什么，展现自己能力。少说与测试无关的内容，尽量展示出自己专业的地方。

## 针对于支付模块，如果接入了支付宝该如何测试

对于普通商城系统而言，如果接入的是支付宝从测试的角度，需要与支付宝对接联测方可完成测试。在这种情况，一般我们不会选择与支付宝的测试团队对接，因为过于麻烦。反而我们会选择接入第三方真实系统进行测试，来模拟我们的过程。即付款时走真支付宝付款，实际上付款是将某账户中的钱转入到另外一个账户，如果两个账户都是自家的则不需要担心金钱损失。

例如押金、支付、充值都可以考虑这种情况，甚至还有短信平台也会接入真实系统。

这种接入第三方真实系统，有一些注意事项：因为涉及到金钱交易，最好与公司申请得到公司的支持，如有必要可使用小额账务完成整个过程。另外，在这种情况涉及到的金钱交易要特别小心，以免造成公司损失。

一般有支付，就会有退款操作。有时候还需要验证退款操作，检查退款到哪里去了，是否原路返回，是否存在手续费扣减。

# 缺陷管理相关

## Bug报告包括哪些内容

开始时间-结束时间，缺陷标题，优先级，前置条件，重现步骤，测试结果，预期结果，测试账号，测试环境测，测试设备，附件，重现频率，执行人&参与人，

缺陷状态

|  |  |  |
| --- | --- | --- |
| 测试人员修改：fixd | 测试人员回归测试通过：closed | 回归测试未通过：Reopen |

## 缺陷处理流程：

1.开发认为是缺陷的处理

    测试人员发现并提交缺陷，由开发人员进行处理，开发人员修改了这个缺陷就会将这个缺陷的状态置为Fixed状态让测试人员进行验证。测试人员对这个已修复的缺陷进行回归测试，如果回归测试通过，则将缺陷状态置为closed，如果回归测试没有通过，则将缺陷状态置为Reopen状态等待开发再次修复，直到修复成功。

2.开发认为不是缺陷的处理

     测试人员发现并提交缺陷，由开发人员进行处理。但是开发人员认为不是缺陷，则将该缺陷的状态置为Reject状态并提交回测试人员。测试人员如果认为确实误报了缺陷，则直接关闭（Closed），如果经过测试、开发沟通认为是bug，则测试人员重新打开（Reopen）让开发人员继续修改，开发人员修复这个缺陷置为Fixed，提交给到测试人员进行回归测试，直到回归测试通过为止。

3.开发认为重复缺陷的处理

     测试人员发现并提交缺陷，由开发人员进行处理。但是开发人员认为是重复缺陷，则将该缺陷状态置为重复缺陷，作为测试人员一定要确认该缺陷是否确实有人处理（获取到重复的缺陷ID），如果确实是同一个缺陷，则将重复的缺陷直接关闭。如果不是同一个缺陷，则重新打开该缺陷，继续跟踪。

4.延迟缺陷的处理

     测试人员发现并提交缺陷，由开发人员进行处理。但是因为项目和时间等因素，某些缺陷无法在项目周期内完成，则需要进行延迟处理（备注：延迟处理的缺陷本身被确定为有效缺陷），对于延迟的缺陷需要经过开发、测试、项目经理、客户代表共同认可方可延迟。对于延迟的缺陷，置状态为Delay（测试人员翻转该状态）到了下一个版本，测试人员就应该把所有Delay状态的缺陷重新置为Reopen状态，让开发人员继续修复。

## 随机缺陷怎么测？

致命：程序的主要功能丧失，死机，闪退，核心业务没有实现等。

严重：次数的功能没有实现，引起部分功能（场景）无法使用，例如金额计算错误，无法删除数据等

一般：基本功能实现，但是边界值错误

轻微：界面的排版错误，系统操作不方便但可以使用

## 随机缺陷怎么测？

1. 先提交到缺陷管理库
2. 对第一次出现的缺陷要截图，尽量回想发现问题的步骤，进行步骤重现。
3. 喊开发人员先开启相关模块的日志，下次出现的时候有日志可查
4. 再次出现随机缺陷，保留现场，喊开发人员来看
5. 实在无法重现，在严重性也很低的时候，可以暂不处理，等到再次遇到该缺陷时修复。

## 我现在有个程序，发现在Windows上运行得很慢，怎么判别是程序存在问题还是软硬件系统存在问题？

参考答案：

1、检查系统是否有中毒的特征；

2、检查软件/硬件的配置是否符合软件的推荐标准；

3、确认当前的系统是否是独立，即没有对外提供什么消耗CPU资源的服务；

4、如果是C/S或者B/S结构的软件，需要检查是不是因为与服务器的连接有问题，或者访问有问题造成的；

5、在系统没有任何负载的情况下，查看性能监视器，确认应用程序对CPU/内存的访问情况。

# 测试用例设计面试题：

## 1.关于验证码的问题：

1.验证码的测试用例你会从哪几个方面考虑？

2.突然收不到验证码，你会从哪几个方面考虑？

**参考答案：**

验证码一般是用来做登录密码，或者一些其他安全性较高的业务功能。验证码的测试是一种典型的匹配类测试，自然也就有正确验证码、错误验证码之分。另外验证码还有一个典型的特征是具有时效性，即2分钟会失效还有一个特征是验证码能够重发，一般设置1分钟重发。

验证码一般是由公司短信平台系统发出，在测试的过程中。我们在自己的业务系统点击获取验证码，再去短信平台查询验证码，所以不存在所谓的收不到验证码，因为在测试系统中，验证码根本就不发送到手机上来的。

如果真收不到验证码，可能有如下情况：

1.短信平台和业务系统环境出问题了，解决环境问题；

2.系统稳定性不够，请求没发出，再重发一次就好了。

3.是否你的业务账号记错了，那属于工作失误。

4.是否你的手机号被别人使用了，产生手机号的冲突。

## 2.新增和修改页面的用例设计和Bug提交

**问题：**

新增页面和修改页面，基本上输入框都一样，那比如同一个输入框的用例设计：

1. 写了新增页面的用例，修改页面对该输入框还有再写一遍用例的必要吗？

2. 执行用例时，新增页面验证了必填项，长度，数据类型，修改页面还要再验证一遍吗？

3. 提交Bug时，新增和修改页面的同一个输入框都出现了Bug，是只提交一个还是新增和修改各提一个。

**参考答案：**

我们写用例最容易落入一个误区，就是为了写用例而写用例。实际上写用例最主要目的是分析系统，如果系统业务复杂，用例分析与设计就很重要，如果很简单的功能不写用例也能够搞定执行。就好像我们做数学题1+2基本上不用思考就知道等于3，如果是546+788，我们甚至可能要列竖式认真算，这也是如果我们遇到简单功能，不用过多写用例就能测好，那么何必写？

想必说到这里，我们的答案已经清晰了，新增的用例写完后，我们完全可以参考新增的用例去执行修改功能，不需要重复写用例了。

当然第二个问题，我就不太能确定了。新能功能和修改功能是否共用一个程序？如果是，那么可以不用测试，如果不是，那么需要测试，要不要测试取决于系统的实现方式。

第三个问题，同样我们需要思考我们提交缺陷的目的所在，我们是要开发能够修改所有的缺陷即可，那么不在乎到底是写在一起还是分开写，只要清晰体现即可。我一般做法是写在一起，说明同样的缺陷在其他具体哪些功能同样存在。

## 3.关于登录功能的测试点：

1.用户名和密码均正确，登录成功

2.用户名错误，登录失败

3.密码错误，登录失败

4.用户名为空，提示报错

5.密码为空，提示报错

6.密码错误3次，账号锁定

7.账号锁定，登录失败

## 4.微信群发红包的测试点：

1.此为拼手气红包，应检查收到的红包金额不均等，是拼手气的。

2.红包如果没有被抢完，24小时后钱会原路退回至发红包者账户。

3.红包已抢完，再抢红包会提示红包已被抢光

4.同一个人不能两次抢同一个红包，发红包者可以抢红包。

5.红包个数支持1~100个，无效取值为0，101个（边界值）

6.红包金额为人均0.01~200元，无效取值为：人均小于0.01，人均大于200元。

7.考虑红包个数和金额的边界组合，最小个数最小金额（1个0.01元），最小个数最大金额（1个200元），最大个数最小金额（100个1元），最大个数最大金额（100个2万元）。

8.使用自定义附言和默认附言发送红包，附言还支持图片附言。

9.如果账户余额不够，支持从银行卡进行扣款。

10.节假日等特殊日期可以发送大于人均200元的红包

## 5.如何测试一个水杯：

关于一个水杯如何测试？这个被认为是测试界最为经验的面试题了，下面是我的回答思路：

对于一个软件的测试，重点是测试的思路以及测试的全面性的体现。

软件测试应该先重点再次重点，对于软件而言重点自然在于功能测试，尔后再非功能测试。

对于功能测试，重点在于核心功能的测试，尔后再其他功能的测试。

所以沿着核心功能-->其他功能-->非功能，整理出来的测试思路为：

1. 我们要进行功能层面的测试

　　　　a)    核心功能的测试（装水）

                i.      能否装常温水 ii.     能否装开水 iii.     能否装冰水

　　　　b)    附加的功能（装别的东西）

                i.      能否装饮料 ii.     能否装硫酸（不能装的那类）       iii.    能否装固体（装饭）

　　　　c)    其他功能

                i.    作为礼品（以前流行金杯银杯不如百姓的口碑，现在流行杯具）

　　2. 说完功能层面的，再来讨论非功能层面的

　　　　a)  性能层面

              i.     耐高温（是否能在微波炉中使用）            ii.    厚度和硬度（多高摔下来不破）

　　　　b)   易用性层面

               i.     是否可以方便的喝到水  ii.    是否很好的手持

iii.   是否防滑                iv.   是否界面美观

　　　　c)   安全性层面

              i.    是否不会割伤嘴巴       ii.    是否盛热水不会烫手（隔热）

　　　　d)    其他层面

　　　　　　　　抛砖引玉，留给大家扩展。

## 6.我手上这支笔，请你根据这支笔设计测试用例

同水杯测试思路

## 7.等价类设计用例的步骤

1. 对输入项进行有效等价类和无效等价类划分，得到等价类表。

2. 设计一个新的用例，使其尽可能多的覆盖尚未覆盖的有效等价类，重复这一步骤，直到有效等价类都被覆盖为止

3. 设计一个新的用例，使其仅覆盖一个尚未覆盖的无效等价类，重复这一步骤，直到所有的无效等价类都被覆盖为止。

## 8.场景分析法设计用例的步骤

1. 分析需求，根据需求规格说明书画出流程图

2. 确定流程图中的基本流和备选流，根据基本流和备选流生成不同的场景

3. 对每一个场景使用若干个数据用例进行覆盖

# 常见技术面试题集：

## 写出你熟悉的10个Linux命令

i（询问） r（递归） f（强制）

clear：清屏 pwd:显示当前工作路径 cd:切换工作路径

vi：打开vi编辑器 :wq! 强制保存文件并退出 /abc 查找文件 n：查找下一个 dd删除整行 yy：复制整行 p 粘贴 i:转到插入模式 esc键：退出插入模式

top：实时显示系统中各个进程的资源占用情况，可以查看cpu和内存的使用情况

su：切换用户 su root是切换到root账号

rm –rf 文件目录：删除文件或文件夹，不提示。 rmdir 目录名称：删除空目录

mv 文件 目标路径：移动或更名。 cp：复制文件或目录

mkdir 目录名称：创建目录。 rmdir 目录名称：删除空目录。 touch：创建文件。

chmod +x file：更改文件权限，增加可执行权限 chmod –R 777 目we'we录：增加全部权限

find / -type f –name 文件名称：查找文件命令。

grep –n password mysql.log：在mysql.log文件中查看password关键字的行

tail –f xx.log：监控日志文件 cat -n：显示文件中的全部内容，显示行号

ps –ef|grep 进程名：查看进程 kill -9 pid：强制杀死一个进程

tar –zxvf xx.tar.gz：解压缩tar.gz文件 unzip xx.zip：解压缩zip文件

yum install httpd：在线安装httpd wget http://mysql.com/mysql8.0.cgi :下载软件

systemctl start httpd:启动httpd服务，stop是停止

head -num 文件名：head显示文件头 tail -num 文件名：tail显示结尾

head -50 stdio.h | tail -20 ：查看中间部分

## 你之前在公司使用Linux命令做什么

我们公司之前测试系统搭建在Linux上，使用Linux搭建和升级测试环境、查看后台日志等。注：Linux安装在物理机上，或者安装在阿里云上，在Linux安装过禅道缺陷管理工具（centos8无坑）。

## Linux查日志操作

在一台Linux服务器上查看下单接口的日志order.log并通过关键字（90018927）定位到该日志范围，请写出Linux操作命令

参考答案： grep “90018927” order.log

## Linux中如何查看进程？如何杀死一个进程？

查看进程：ps -ef|grep 进程名称

如何杀

1. 如果进行是服务，可以通过systemctl stop 进程名称完成，例如：systemctl stop mysqld
2. 如果进行是sh文件，可以运行这个sh文件，例如： ./stop-jira.sh 停止jira
3. 可以通过kill -9 pid来完成强制杀死进程，例如：kill -9 2144

## Linux版本

常见的Linux版本有：CentOS7或8，Ubuntu18等

## 远程终端工具：

远程连接工具：Xshell、SecureCRT、文件上传：FTP

## 使用数据库进行哪些工作？

可以在数据库中捞取合适的测试数据进行测试（比例最高的）

证数据一致性，编写过复杂的查询语句，多表关联等，验证数据库正确性，对于前台输入的数据和数据库增加的记录进行一致性对比。

更新数据，提高测试效率。制造一些特殊数据，便于测试的正常进行。

获取数据库中的中间结果再做进一步操作，因为通过SQL直接操作数据库，那么可以通过SQL语句来完成你所想要的任意操作

进行缺陷定位，通过数据库手段定位缺陷的原因。特别是对于生产缺陷、用户提交的问题进行定位。

## 数据库中删除命令有哪些？他们之间的区别？

删除命令有：delete from 表名称 WHERE 子句和truncate table 表名称

* delete from 可以接WHERE子句，可以进行部分删除，也可以不接WHERE子句进行全表删除。
* truncate table 只能进行全表删除。
* delete from是一行一行删除的，所以删除速度慢。truncate table是页删除，删除速度快。
* delete from 删除可以通过rollback回退，truncate table不可以。

## 数据库基础知识：

数据库的增删改查、分组查询（聚合函数）、排序、子查询、表连接查询、多表查询、表操作等。

数据插入数据的语法：

INSERT INTO 表名称 VALUES(‘值1’,’值2’,’值3’) ;

更新数据的语法：

1.标准语法：

UPDATE 表 SET 字段=新值 WHERE 筛选条件 ;

2.更新为默认值和空值

UPDATE 表 SET 字段1=DEFAULT,字段2=NULL WHERE 筛选条件

删除数据的语法：

1. 删除部分数据的语法：

DELETE FROM 表 WHERE 筛选条件

1. 删除全部数据的语法：

DELETE FROM 表名称 ;

TRUNCATE TABLE 表名称; 截断表，也可以删除全部数据

TRUNCATE和DELETE的区别：

1. DELETE可以删除部分数据，也可以删除全部数据。TRUNCATE只能删除全部数据
2. DELETE删除后，被AUTO\_INCREMENT自增长的字段值不被清零，而TRUNCATE后自增长字段会清零
3. TRUNCATE是页删除，删除速度快。DELETE是行删除，删除速度慢。

## 数据库查询语法：

单表查询

1. 查询的标准语法

SELECT \* FROM 表

1. 查询时使用表达式 IFNULL如果字段值为空则用0表示，如果非空则为原来数据

SELECT sno,IFNULL(score,0) FROM 表 ;

1. 查询并去重复 DISTINCT

SELECT DISTINCT job,city FROM 表 ;

1. 查询只显示前5行 LIMIT（Mysql独有）

SELECT \* FROM 表 LIMIT 0,5

条件筛选语句：

1. 等于筛选

SELECT \* FROM 表 WHERE sname=’admin’ ;

1. 不等于筛选 > >= < <= != <>

SELECT \* FROM 表 WHERE score>=60 ;

1. 模糊查询

SELECT \* FROM 表 WHERE sname LIKE ‘%A%’ ; # NOT LIKE

1. 空值处理

SELECT \* FROM 表 WHERE score IS NULL ; #IS NOT NULL

1. 条件组合（AND OR）

SELECT \* FROM 表 WHERE sname LIKE ’%A%’ AND score>=60 ;

SELECT \* FROM 表 WHERE sname=’admin’ OR score is null ;

1. 条件区间 BETWEEN AND

SELECT \* FROM 表 WHERE score BETWEEN 60 AND 80 ;

1. 在集合中

SELECT \* FROM 表 WHERE sname IN (‘admin’,’tester’,’qianli’) ;

分组查询

1. 分组函数（聚合函数）

COUNT：统计数量 MAX：求最大值 MIN：求最小值 AVG：求平均值 SUM：求和

1. 分组函数语法

SELECT 组名称（也叫分组项）,分组表达式 FROM 表 WHERE 筛选条件

GROUP BY 分组项 HAVING 二次筛选

1. HAVING二次筛选

WHERE一次筛选用在普通字段的过滤上

HAVING二次筛选用在对聚合函数的筛选上

多表关联查询

1. 等值连接查询（内连接）

SELECT a.\*,b.\* FROM a,b WHERE a.sno=b.sno and a.score>60 ;

SELECT a.\*,b.\* FROM a join b on a.sno=b.sno WHERE a.score>60 ;

1. 自身表连接

SELECT a.\*,b.\* FROM student a,student b WHERE a.sno=b.tno and a.score>60 ;

1. 非等值连接

SELECT a.\*,b.\* FROM a join b on a.score between b.min and b.max WHERE a.score>60 ;

1. 左右连接

SELECT a.\*,b.\* FROM a **left join** b on a.sno=b.sno WHERE a.score>60 ;

SELECT a.\*,b.\* FROM a **right join** b on a.sno=b.sno WHERE a.score>60 ;

子查询：

1. 子查询的含义：就是查询中嵌套查询
2. 子查询有三种表现方式：
   1. 子查询在FROM 子句中
   2. 子查询是一个数据，在WHERE子句中直接可使用 = (子查询) 即可
   3. 子查询是多个数据，在WHERE子句中需要使用 列 IN (子查询)

## 左连接和右连接的区别

1. 他们的相同点：左连接和右连接同属于外连接，在具体实现中可以左右连接互换
2. 左连接是以左边表为基准表，右表接左表，如果右表没有数据对接左表则用NULL补充
3. 左连接的关键字是LEFT JOIN 或LEFT OUTER JOIN，右连接的关键字是：RIGHT JOIN或RIGHT OUTER JOIN
4. 在进行左右连接的时候，需要明确查询时的主语（主表）是谁

## 浏览器中输入一个网址他的运行过程是怎么样的？

http://192.168.1.129:8080/website/install/step1.php

* 我要查看的网页是：/website/install/step1.php
* 它的WEB服务器端口（管家编号）是：8080
* 这个网页所属服务器（它在哪台电脑）是：192.168.1.129
* 协议是http

## WEB（B/S）的前端与后台

**前端：**

HTML：超文本标记语言，用于生成静态网页，例如输入框，按钮等。

CSS：层叠样式表，用于设置页面样式，例如输入框的大小，按钮的位置等。

JS：JavaScript，一种脚本语言，用于生成动态页面，进行简单的页面数据效验（简单密码提示等）

**后台：**

WEB服务器：Apache、Tomcat、Nginx、Weblogic、IIS等。

数据库服务器：Oracle、Mysql、SQL Server、MongoDB等

## OSI七层模型从低到高依次为

这个题目是考核计算机基础的，能够回答出来即可，不用深究。

**物理层：**光纤，双绞线（即普通网线）、网卡、集线器等

**数据链路层：**网桥、交换机

**网络层：**网关、路由器。IP协议在网络层

**传输层：**TCP和UDP协议在传输层

**会话层：**在应用层建立会话

**表示层：**对应用数据进行压缩和加密

**应用层：**HTTP协议、HTTPS协议属于应用层协议

## TCP和UDP的区别

1、TCP面向连接（传输数据之前需要三次握手）;UDP是无连接的，即发送数据之前不需要建立连接

2、TCP提供可靠的服务;UDP尽最大努力交付，即不保证可靠交付

3、TCP传输速度慢（单位时间发送的有效数据更少）,UDP传输速度快

4、TCP适合传输少量数据（数据质量要求高）;UDP适合传输大量数据（数据质量要求相对低）

腾讯QQ使用的是UDP协议进行传输

# 业务测试题集

## 说说做了2年测试工作的一个体验

我觉得编写文档的能力非常重要（体现在用例、缺陷和报告）、要保持一颗不断提高高级技术的心（数据库、接口、自动化、性能等）、一定要多问产品，有任何有疑问或者不理解的地方要向产品确认清楚，还有一些产品需求里没有涉及到的特殊情况，也要和产品确认

## 测试用例怎样进行评审

用例设计的结构安排是否清晰、合理、优先级安排是否合理

【重点】是否覆盖测试需求的所有功能点

【重点】用例是否具有很好的执行性（站在测试角度如果用例不好执行可以提出期望的协助）

是否已经删除了冗余的测试用例、是否包含充分的测试用例

是否从用户层面来设计用户使用场景和使用流程的测试

## 项目延期会怎样

1. 在项目延期之前，提出风险。
2. 对延期做出努力，尽力减少延期
3. 分析导致延期的原因，努力避免下次因为同样原因延期

## 项目需要提前上线怎么办？

1. 提前肯定是有原因的，对合理的原因予以支持
2. 工期被压缩了，提出风险
3. 需调整测试策略，在上线前完成核心工作

## 项目上线标准

上线前先制定好上线标准：测试用例的覆盖率达到100%，如有测试用例没有执行，一定要写明原因，确定它影响其他功能的机率。严重的Bug数和一些普通的功能缺陷为0，如果是轻微的界面问题，开发人员没有时间修改，需要写明原因，留下一个版本上线后再测试。

## 你认为测试最重要的是什么

【提高质量】尽可能的找出软件的错误，本质上提高用户对软件的满意度

【提高效率】如何用最短的时间、最少的测试用例达到最全面的测试效果

【做必要的事情】如何明确确定需求范围，确定测试策略

## 为什么在团队中要有测试0

【专业性】测试人员可以发现更多的缺陷

【高效】测试人员的专业性可以让测试的效率变得更高

【全面】测试人员负责整个系统的测试，对整个系统的需求和业务都非常清晰，能够更好的发现整个系统的缺陷，以及使用更全面的测试策略

## 认为测试人员需要具备哪些素质

技术的：用例设计方法，需求分析能力，项目管理能力，自动化测试，接口测试技术，了解网络及协议，数据库知识等

非技术的：我认为测试人员应具备一定的协调能力，因为测试人员要经常与开发接触处理一些问题，如果处理不好的话会引起一些冲突，这样的话工作上就会做不好。还有测试人员要有一定的耐心，有时候做的测试很枯燥乏味。除了要有耐心之外还要细心，不放过每一个可能的错误细节。

## 测试人员在需求分析阶段要做什么

与开发、客户在一起评审，对需求文档进行评审。找出需求中错误的地方，不明确的地方，如果测试人员没有看懂需求的内容，也可以在分析阶段提出来 。

## 工作中遇到了问题你会怎么解决，百度还是问同事？

这是一个送命题，如果你说百度，面试官可以问为何不问同事，因为问同事效率高。如果说问同事，面试官又可以说老问同事，同事可能会烦的。

解决建议：对问题进行分类，不同的问题使用不同的方案

1. 如果是技术问题，直接百度，百度可以快速的找到问题的答案。一般我找cnblogs（博客园）和CSDN的答案。
2. 如果是带分析的技术问题，会先思考可能原因，仍然会进行百度。如果百度了不能解决我的问题，会求助同事的帮助。
3. 如果是与公司项目和业务相关的问题，预计百度不到，则会直接问同事，寻求同事的帮助。

## 项目中最困难的一件事情

1. 需求不明确，一页纸的需求。解决方案：逼得我亲自写需求了，嘿，效果还可以。
2. 还有就是需求变更，不过引入了敏捷开发，项目周期短缩后需求变更就少多了

# 敏捷开发测试相关知识

## 没有任何说明书，如何进行测试

1. 和开发、产品进行沟通，掌握软件的特性并一一记录下来，作为测试的依据
2. 查看同类产品的特性和需求，并作为测试的参考
3. 软件的实现和逻辑、规则都需要符合用户的正常使用行为，可以站在用户角度进行分析与判断
4. 根据测试人员过往的经验进行分析

## 用过什么测试管理工具，流程是什么

我所熟悉的是禅道，jira等，我使用过的是禅道，之前我们公司用来做测试管理，先新建产品—项目—项目的功能模块，然后把项目组人员配置到该项目，使用导出测试用例模板，编写评审好测试用例，再导入禅道管理，执行测试用例，提交缺陷到禅道并跟踪。

## 你们公司的开发模式是什么，具体项目如何执行

我们公司之前采用敏捷开发模式，下图是敏捷开发的看板任务。

## User Stroy用户故事

像用户讲故事一样，有点像在讲测试点。作为一个XX用户，我期望实现一个XX功能，以达到XX目的

该需求一般比较短

## 你知道A/B测试和灰度发布吗

AB测试是为Web或App界面或流程制作两个（A/B）或多个（A/B/n）版本，在同一时间维度，分别让组成成分相同（相似）的访客群组（目标人群）随机的访问这些版本，收集各群组的用户体验数据和业务数据，最后分析、评估出最好版本，正式采用。

灰度发布（又名金丝雀发布）是指在黑与白之间，能够平滑过渡的一种发布方式。在其上可以进行A/B testing，即让一部分用户继续用产品特性A，一部分用户开始用产品特性B，如果用户对B没有什么反对意见，那么逐步扩大范围，把所有用户都迁移到B上面来。灰度发布可以保证整体系统的稳定，在初始灰度的时候就可以发现、调整问题，以保证其影响度。

# 人事面试题集

## 你三到五年的职业规划是怎样的

这个问题是通过侧面考察求职者的上进心和稳定性（重点），所以防止进坑是重点。参考答案：

我想先做好手上的工作，做好后会考虑能够做一些提升工作效率的事情，比如一些工作过程的优化，技术提升和团队管理工作。

如果继续询问技术提升会如何提升则回答：技术的提升看公司需要用到哪些方面吧，会根据公司实际需要进行使用。不过目前测试技术比较火一点的是自动化测试和接口测试两部分，我会考虑从这两方面提升着手。

## 你觉得你的优点和缺点是什么

这个问题题目看起来很简单，实际上不能给自己挖坑：

优点：可以回答你是计算机专业的，技术还可以。对测试的各方面都有所涉及，知识面也比较全面。

缺点你可以说不算很自信，有时候（不是总是）会担心自己工作没做好。到项目上线的时候，会睡不着觉，失眠。

## 事业和家庭存在冲突怎么处理

事业和家庭，当然是事业为重。家庭不会影响到事业的，现在都是双职工家庭了，女性承担工作被社会接纳度很高的。再可以说，自己很享受工作。因为没有工作，会焦虑（没收入）。然后也不知道自己能做什么，基本上是废的。反而有工作了，会有种充实感。

## 面对不合理的公司制度，你怎么办？

1. 公司制度总有其原因，管理层考虑和员工考虑会不一样。
2. 如果制度有问题，我会把它反馈给上级领导，让他来决策

## 你还有什么问题吗？

1. 询问项目是做什么产品的，询问现有测试团队的规模
2. 一般面试官问这个问题表示面试进入尾声了，也可以说没什么问题

## 如果你做的一项工作受到上级领导的表扬，但你的直接领导却说是他做的，你怎么办

不会和领导去争功劳，通过领导表扬确信了自己做了正确的事情。

## 你会怎么增加测试和开发的之间的友好关系

我不会想着增加测试和开发之间的友好关系，咱们的职责是把工作做好，不是来搞关系的。测试人员通过自己的工作能力，也会让产品质量变得更好，也会得到开发的认可。如果测试和开发关系很好，但找不到缺陷，上线后反而有很多缺陷，这样反而把开发给害惨了。

## 面对领导不平等分配工作，你是如何处理的？

每个人的工作时间都是差不多的，要加班也是大家都加班，领导不会针对某一个人的。很多不公平是自己觉得的不公平，如果调整心态，认为给分配的事情多一点是得到了挑战困难的机会，事情少一点也应该把事情做好，那么心里就不会去想着不平等的了。

## 你喜欢在什么氛围的公司里上班？

这个问题看起来很简单，一般人会说喜欢在和睦、轻松的公司上班。其实这个答案并不好，因为多数公司并不轻松（加班的）。

如果是女孩子，可以说：喜欢一个不抽烟的工作环境。如果是男生，可以说喜欢女生多一点的工作环境。这样的回答可以让面试氛围轻松一点，也不掉坑。

也可以说，平时主要心思都是工作，没怎么研究公司有哪些类型的氛围，其实什么样的氛围都无所谓。IT公司是一群高学历的年轻人团队，氛围一般都还可以的。

# 业务测试题集

## 没有任何说明书，如何进行测试

1. 和开发、产品进行沟通，掌握软件的特性并一一记录下来，作为测试的依据
2. 查看同类产品的特性和需求，并作为测试的参考
3. 软件的实现和逻辑、规则都需要符合用户的正常使用行为，可以站在用户角度进行分析与判断
4. 根据测试人员过往的经验进行分析

## 项目中有没有遇到测试来说无法解决的问题？

1. 项目第一次上线部署时发现了一些问题，经过开发人员和测试人员的共同努力，将这一问题成功解决。
2. 某个测试工具不会使用，经过了一些培训之后能够熟悉的使用这些工具
3. 其他

## 关于测试人员考核

正常的考勤、对工作业绩的考核（具体：用例考核、缺陷考核、文档考核，是否按计划完成了工作）、有没有遗漏测试和产生线上缺陷，有没有突出贡献。

## 说说做了2年测试工作的一个体验

我觉得编写文档的能力非常重要（体现在用例、缺陷和报告）、要保持一颗不断提高高级技术的心（数据库、接口、自动化、性能等）、一定要多问产品，有任何有疑问或者不理解的地方要向产品确认清楚，还有一些产品需求里没有涉及到的特殊情况，也要和产品确认

## 测试用例怎样进行评审

用例设计的结构安排是否清晰、合理、优先级安排是否合理

【重点】是否覆盖测试需求的所有功能点

【重点】用例是否具有很好的执行性（站在测试角度如果用例不好执行可以提出期望的协助）

是否已经删除了冗余的测试用例、是否包含充分的测试用例

是否从用户层面来设计用户使用场景和使用流程的测试

是否简洁、利用性强

## 项目延期会怎样

1. 在项目延期之前，提出风险。
2. 对延期做出努力，尽力减少延期
3. 分析导致延期的原因，努力避免下次因为同样原因延期

## 项目需要提前上线怎么办？

1. 提前肯定是有原因的，对合理的原因予以支持
2. 工期被压缩了，提出风险
3. 需调整测试策略，在上线前完成核心工作

## 项目上线标准

上线前先制定好上线标准：测试用例的覆盖率达到100%，如有测试用例没有执行，一定要写明原因，确定它影响其他功能的机率。严重的Bug数和一些普通的功能缺陷为0，如果是轻微的界面问题，开发人员没有时间修改，需要写明原因，留下一个版本上线后再测试。

## 日常工作内容是什么？

负责\*\*系统的测试用例编写，执行，Bug提交以及项目总结报告的编写。负责测试管理工具的管理，做文档的评审，做客户培训。也可以将测试流程中的活动依次给面试官进行介绍

## 结合你以前的学习和工作经验，你认为如何做好测试工作。

根据我以前的工作经验，我认为做好工作首先要有一个好的沟通，只有沟通无障碍了，才会有好的协作，才会有更好的效率，再一个就是技术一定要过关，做测试要有足够的耐心和良好的工作习惯，不懂就问，实时与同事沟通这样的话才能做好测试工作。

另外：参考前面的第一题，如何提高测试的质量

## 根据以前的工作经验，描述一下软件开发、测试过程，有哪些角色，你做什么

有项目经理、产品经理，开发人员，测试人员。我负责测试工作，全系统的功能测试等。

## 你认为测试最重要的是什么

【提高质量】尽可能的找出软件的错误，本质上提高用户对软件的满意度

【提高效率】如何用最短的时间、最少的测试用例达到最全面的测试效果

【做必要的事情】如何明确确定需求范围，确定测试策略

## 为什么在团队中要有测试

【专业性】测试人员可以发现更多的缺陷

【高效】测试人员的专业性可以让测试的效率变得更高

【全面】测试人员负责整个系统的测试，对整个系统的需求和业务都非常清晰，能够更好的发现整个系统的缺陷，以及使用更全面的测试策略

## 认为测试人员需要具备哪些素质

技术的：用例设计方法，需求分析能力，项目管理能力，自动化测试，接口测试技术，了解网络及协议，数据库知识等

非技术的：我认为测试人员应具备一定的协调能力，因为测试人员要经常与开发接触处理一些问题，如果处理不好的话会引起一些冲突，这样的话工作上就会做不好。还有测试人员要有一定的耐心，有时候做的测试很枯燥乏味。除了要有耐心之外还要细心，不放过每一个可能的错误细节。

## 你为什么做测试这一行

首先我掌握较为全面的测试知识，虽然说我的测试技术还不是很纯熟，但是我觉得我还是可以胜任软件测试这个工作的。因为做软件测试不仅要求技术好，还要有一定的沟通力，耐心，细心等外在因素。综合起来看我认为我是胜任这个工作的。

一般回归测试策略是验证修改的位置以及其周边模块是否存在错误

## 测试人员在需求分析阶段要做什么

与开发、客户在一起评审，对需求文档进行评审。找出需求中错误的地方，不明确的地方，如果测试人员没有看懂需求的内容，也可以在分析阶段提出来 。

## 黑盒测试方法有哪几种

等价类覆盖法，边界值分析法，错误推断法，判定表驱动法，功能图分析法（正交实验），场景法等

## 测试环境找搭建和升级，你之前在公司是如何做的

**完全升级：**根据开发人员给的升级文档，查看代码包、数据库脚本包是否齐备，然后把测试环境的数据库、被测系统代码包先备份然后删除，然后根据开发最新的文件和代码进行升级部署操作。升级之后做个基本功能验证（冒烟测试），如果验收失败，测试环境按之前的备份包回滚。

**增量升级：**和完全升级类似，不需要氢测试环境的数据库、被测系统代码包先备份然后删除，只需要备份，然后覆盖更新的代码和执行数据库脚本即可。

## 冒烟测试的概念是什么，在公司实际如何做的

把开发提交的新版本测试包主要功能验证一次，通过就进行系统测试，没有通过就打回给开发人员，这就是冒烟测试。我们公司之前测试部门有一份《开发提交清单》上面包含主要功能验证，开发提交代码和文档规范，如果不达标，就打回给开发。

## 工作中遇到了问题你会怎么解决，百度还是问同事？

这是一个送命题，如果你说百度，面试官可以问为何不问同事，因为问同事效率高。如果说问同事，面试官又可以说老问同事，同事可能会烦的。

解决建议：对问题进行分类，不同的问题使用不同的方案

1. 如果是技术问题，直接百度，百度可以快速的找到问题的答案。一般我找cnblogs（博客园）和CSDN的答案。
2. 如果是带分析的技术问题，会先思考可能原因，仍然会进行百度。如果百度了不能解决我的问题，会求助同事的帮助。
3. 如果是与公司项目和业务相关的问题，预计百度不到，则会直接问同事，寻求同事的帮助。

# Python自动化测试技术

## 手工测试能代替自动化测试吗？

原因一：自动化不适合实现某些特别复杂的场景，其实现成本特别高。例如注册，每次成功注册要更换数据。密码连续错误3次账号锁定，这些偏复杂。甚至还有更复杂的，不适合用自动化实现的场景。

原因二：一些偏主观的测试，比如易用性测试，UI测试不适合用自动化测试实现。

原因三：如果是短期项目，测完就结束了，那么自动化成本也会很高。

原因四：如果项目稳定性不够，自动化的维护成本会很大。

不过接口自动化测试的应用场景比较好，因为接口自动化，不测偏主观的界面，一般接口是比较稳定的。同时实现起来也比较容易，balabala

## 你主要学了什么编程语言？我的是Python

python元组是('a','b')，其特点是一旦元素确定了，不可更改内容也不可更改元素的个数。

list列表是['a','b']，其特别是值可变，元素个数也可变。索引是从0开始的自然数

字典是{'name':’qianli’,'age':18}，由key:value组成

## Python的基础数据类型

使用type()可以查看数据的类型

1. 数值类型
   1. 整数（init）
   2. 浮点数（float）
   3. 布尔值（bool）：布尔值有且只有两个值：True和False
   4. 空值：None
2. 字符串类型
3. 复合数据类型
   1. list列表类型：[1,2,3,4]
   2. tuple元组类型：(‘a’,1,1.5) 元组一旦创建便不再能够修改
   3. set集合类型：{‘a’,1,5,None} 集合中的元素不能重复
   4. dict字典：{‘name’:’admin’,’age’:18} 字典具有key:value的特征

## 字符串常用的方法

分割字符串：str.split(‘分割标志’) 分割后返回列表类型

字符串转大写：str.upper()，转小写：str.lower()

去掉两端的空格：str.strip()

获取字符串的长度：len(str)

## 安装模块的方法

命令安装：命令行窗口输入 pip install 模块名

## 常用模块-time模块

获取当前时间： time.ctime()

time.strftime(‘%Y-%m-%d %H:%M:%S’) 自定义格式，位置与连接符号可自定义

## Selenium中如何保证操作元素的成功率，也就是说如何保证我点击的元素一定是可以点击的？

添加元素智能等待时间，driver.implicitly\_wait(30)

try方式进行id,name,class,xpath,css selector不同方法进行定位，如果第一种失败可以自动尝试第二种。

## 用例在运行过程中经常会出现不稳定的情况，也就是说这次可以通过，下次就没办法通过了，如何去提升用例的稳定性

time.sleep()

driver.implicitly\_wait(30)

## 你的自动化用例的执行策略是怎样的

手工执行

定时任务触发执行（周末或晚上），可以使用windows计划任务、Jenkins来进行定时任务

Jenkins自动构建执行（Jenkins定时去比较当前版本与svn上的版本是否一致，如何不一致则更新，一致则不更新）

## 什么是持续集成

持续集成是源于极限编程（XP），是一种软件实践，软件开发过程中集成步骤是一个漫长并且无法预测的过程。集成过程中可能会爆发大量的问题，因此集成过程需要尽可能小而多，实际上持续集成讲的是不断去做软件的集成工作。

## 持续集成如何做

最简单的形式是包括一个监控版本控制（SVN等）变化的工具，当变化被发觉时，这个工具可以自动的编译并测试你的应用。

维护一个单一的代码库，使构建自动化，使构建自测试。每人每天都向主线提交代码，每次提交都应在集成机上进行构建，快速构建，使任何人都轻易获得可执行文件，人人都能看到正在发生什么，自动化部署。

## 自动化测试时候是不是需要连接数据库进行数据校验

一般UI自动化不需要，接口自动化测试可能会需要（实际上一般也不需要）

## id,name,xpath,css selector这些属性，你最偏爱哪一种，为什么

优先id，name进行定位，简单高效，其次是css

xpath效率最低，但定位的效果最好，因为使用层级结构来定位，一般其他方法搞不定的时候用xpath都能够定位到元素。

## 如何去定位页面上动态加载、变化的元素

触发动态加载元素的事件，直到动态元素出现，进行定位。

xpath或css通过同级、父级、子级进行定位。

## 点击链接后，Selenium是否会自动等待该页面加载完毕？

会的

## webdriver client的原理是什么

webdriver是按照Server-Client的经典设计模式设计的。

Server端就是remote server，可以是任意的浏览器，当我们的脚本启动浏览器后，该浏览器就是remote server，它的职责就是等待client发送请求并做出响应。

client端简单来说就是我们的测试代码，我们测试代码中的一些行为，比如打开浏览器，跳转到特定的url等操作是以http请求的方式发送给被测浏览器，也就是remote server.

remote server接受请求，并执行相应操作，并在response中返回执行状态，返回值等信息

## webdriver的工作流程

当测试脚本启动了firefox的时候，selenium-webdriver会首先在新线程中启动firefox浏览器。如果测试脚本指定了firefox的profile，那么就以该profile启动，否则的话就新启1个profile并启动firefox

firefox一般以-no-remote的方法启动，启动后selenium-webdriver会将firefox绑定到特定的客户端（也就是测试脚本）创建1个session，在该session中通过http请求向remote server发送restful的请求，remote server解析请求，完成相应操作并返回response；客户端接受response，并分析其返回值以决定是转到第3步还是结束脚本。

## webdriver的协议是什么

The WebDriver Wire Protocol

## 什么是Page Object设计模式

Page Object模式是Selenium中的一种测试设计模式，主要是将每一个页面设计为一个Class，其中包含页面中需要测试的元素（按钮，输入框，标题等）。这样在selenium测试页面中可以通过调用页面类来获取页面元素，这样巧妙的避免了当页面元素id或位置变化时，需要修改测试页面代码的情况。

当页面元素id变化时，只需要更改测试页Class中页面的属性即可。获取页面中元素的属性可以通过id,class或xpath获取，在id唯一的情况下可以使用id获取页面元素，否则可以使用xpath定位页面元素。

## 怎样去选择一个下拉框中的value=xx的option

二次定位或使用Select类（Selenium自带）

## 什么是断言

断言的英文是assertion，断言检查的英文是assertion checking。断言是指定一个程序必须已经存在的状态的一个逻辑表达式或一组程序变量在程序执行期间的某个点上必须满足的条件。

## 自动化测试用例从哪里来

手工用例中抽取，可以参考自动化用例的执行策略

# 接口测试技术专题

## 接口测试流程是什么

接口测试是测试系统组件间接口的一种测试。接口测试主要用于检测外部系统与系统之间以及内部各个子系统之间的交互点。测试的重点是检查数据的交换，传递和控制管理过程，以及系统间的相互逻辑依赖关系等。

接口测试也是属于功能测试，所以跟我们以往的功能测试流程并没有太大区别，测试流程依旧是：

1、评审测试接口文档（需求文档）得到其接口地址、入参和出参（预期结果）等信息

2、根据接口文档编写测试用例（用例编写完全可以按照以往规则来编写，例如等价类划分，边界值等设计方法）

3、设计接口脚本，使用Postman或JMeter填入接口请求、测试用例以及断言，实现完整的接口测试脚本。

4、执行测试，查看不同的参数请求，接口的返回的数据是否达到预期。

## 接口测试的测试重点是什么

**一、功能测试：**

\* 接口的功能是否正确实现了

\* 错误码测试：通用的错误码与业务错误码是否能够清晰的说明调用问题，错误码是否能够尽可能的全的覆盖所有的情况

\* 返回值测试：返回值除了内容需要是正确的，还需要类型也是正确的，保证调用方拿到这些参数能够正确的解析

\* 参数边界值、等价类测试、 json格式测试：通常我们的接口一般设计的都是传递json串，那么就需要去测试 如果传递非json的情况，这时候程序会不会正确的处理，返回相应的 error code

\* 默认值测试：很多情况一些非必填的参数会有默认值，比如说一个查询的接口，参数count为返回查询的结果数量， 默认为10，那么就应该有一条case来测试，当然前置条件是数据库里面必须要存在这样的数据超过10条。

**二、逻辑业务：**

\* 是否有依赖业务，比如查看订单，是需要用户首先登录的，所以肯定要保证登录了或有相应的cookie

\* 业务逻辑测试：传递正确的参数，接口对数据库进行查询的操作，需要去验证数据库查询是否正确，接口对数据库进行 增删改的操作，也需要看数据库是否同步进行了这些操作

**三、异常测试：**

异常分为两类，参数异常和数据异常

1、参数异常：

\* 关键字参数：将参数写为开发语言中的关键字

\* 参数为空：比如去掉了username参数

\* 多或少参数：多或者少参数的验证，现在还不确定如果一个接口多了参数如果没有报错是否是合理的，或者是否需要优化，因为就目前开发给予的答案是，一般不对接口多了参数的处理

\* 错误参数：比如将username参数写为了user等看是否能返回相应的error code

2、数据异常：

\* 关键字数据：将参数的值填为开发语言中的关键字

\* 数据为空：将参数的额值填为空

\* 长度不一致：因为数据库中每个字段都设置有字段长度，填写不符合的长度进行验证

\* 错误数据：就是将参数的值任意填写，或填写不存在的数值

\* 异常类型测试：比如count参数，这个参数的类型一定是可以转换为int类型的，这时候我们需要测试如果传的一些不可以 转换为int类型值来测试代码是否加入判断

## 为什么开展接口测试

1. 接口测试属于集成测试，测试介入越早越好，越在项目早期发现问题，其修复问题的成本越低
2. 接口测试非常快速，UI自动化执行一个测试用例10S，接口测试用例执行话，需要时间是毫秒级

## 接口测试过程中遇到什么问题

1. 开发设计文档出现问题
2. 接口地址错误
3. 串联接口传值的问题（例：本来要接收一个int数据，结果传过来String类型的数据导致接口不能正确处理）

## cookie和Session什么意思，有什么区别

Cooke：一种客户端机制，浏览器第一次访问服务器时，由服务器返回，然后保存在浏览器中。有效期分为临时和非临时两种，临时cookie会在关闭浏览器后自动删除。cookie的作用是为了保存用户的偏好信息， cookie的大小一般不超过3K

Session：一种服务器机制，浏览器每次访问服务器时，服务器都会返回新的sessionid，sessionid保存在浏览器，session数据保存在服务器。默认30分钟有效。sesson的作用是为了验证用户身份。

**cookie和session的区别：**

1. 数据存放的位置不同：Cookie数据存放在客户的浏览器上，而Session数据存放在服务器上
2. 安全程度不同：Cookie相对不安全，Session相对安全一些
3. 性能使用程度不同：Session会在一定时间内保存在服务器上，当访问增多，会比较占用服务器的资源
4. 数据存储大小不同：单个Cookie保存的数据不能超过4K，而Session对其没有限制。

## GET和POST的区别

get和post是HTTP与服务器交互的方式，说到方式，其实总共有四种：put，delete，post，get。他们的作用分别是对服务器资源的增，删，改，查。在HTTP协议中使用得最为广泛的是GET和POST请求，get是获取数据，post是修改数据。

get把请求的数据放在url上，即HTTP协议头上，其格式为：以?分割URL和传输数据，参数之间以&相连。post把数据放在HTTP的包体内（requrest body），所以POST请求有请求体，而GET请求没有请求体。

GET在浏览器回退时是无害的，POST会再次提交请求。

GET请求只能进行url编码，而POST支持多种编码方式（有四种）。

GET请求参数会被完整保留在浏览器历史记录里，而POST中的参数不会被保留。理论上POST请求更安全，GET请求相对不安全。

## 常见的 POST 提交数据方式

application/x-www-form-urlencoded

multipart/form-data

application/json

text/xml

## HTTP状态码

HTTP状态码2xx,3xx,4xx,5xx分别是什么意思？这个是最基本的了，这个得熟练掌握，如果这个状态码都分不清，基本功就很弱了，印象分会大打折扣！

200 请求已成功，请求所希望的响应头或数据体将随此响应返回。

302 （临时移动） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。

303 （查看其他位置） 请求者应当对不同的位置使用单独的 GET 请求来检索响应时，服务器返回此代码。

304 （未修改） 自从上次请求后，请求的网页未修改过。 服务器返回此响应时，不会返回网页内容。

307 （临时重定向） 服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。

403 服务器已经理解请求，但是拒绝执行它。与401响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交

404 请求失败，请求所希望得到的资源未被在服务器上发现

500 服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器的程序码出错时出现。

501 服务器不支持当前请求所需要的某个功能。当服务器无法识别请求的方法，并且无法支持其对任何资源的请求。

502 作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。

503 由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。

## HTTPS和HTTP的区别主要如下：

Https：是以安全为目标的Http通道，是Http的安全版。Https的安全基础是SSL。

1. http是超文本传输协议，信息是明文传输，https则是具有安全性的ssl加密传输协议。所以HTTPS协议更安全
2. HTTP协议进行数据传输比HTTPS效率高。
3. http和https使用的是完全不同的连接方式，默认端口也不一样，前者是80，后者是443

## HTTP请求报文与响应报文格式

请求报文包含三部分：

a、请求行：包含请求方法、URI、HTTP版本信息

b、请求头部（headers）字段

c、请求内容实体(body)

响应报文包含三部分：

a、状态行：包含HTTP版本、状态码、状态码的原因短语

b、响应头部（headers）字段

c、响应内容(body)实体

# 接口测试常见面试题

## JMeter接口测试流程：

1. 创建线程组作为接口测试的被测功能名称
   1. 设置线程数为1，循环次数为永远
2. 测试步骤的实现
   1. 在取样器中选择添加HTTP请求
      1. 按照HTTP协议特征填写相关要素。需要填写协议、服务器名称或IP、端口号、HTTP请求方法、路径以及消息体数据（请求Body）
   2. 如果请求之间存在关联，则需要做关联操作
      1. 关联的意义：后一个请求的请求参数值需要用到前面请求的响应数据
      2. 关联的操作：
         1. 在前一个请求添加后置处理器🡪正则表达式提取器
         2. 正则表达式提取器名称填写参数名称、正则表达式需要提取的部分使用(.+?)替代，保存左右边界，模板为：$1$ 匹配数字为：1
         3. 在后一个请求的需关联的数据修改为${正则提供器参数名称}
   3. 创建查看结果树，验证请求是否正确
3. 测试数据的实现
   1. 数据用例的设计
      1. 使用等价类、边界值等方法对入参取值进行组合，得到csv数据用例表
      2. csv数据用例表中包含：用例标题、入参、预期结果三项
   2. CSV数据文件设置
      1. 文件名设置中导入csv文件，获取csv的路径
      2. 设置遇到文件结束符不再循环、遇到文件结束符停止线程
   3. 参数化设置
      1. 将HTTP请求中的待参数化的数值设置为${参数}
4. 断言实现
   1. 创建响应断言或JsonPath断言（二者任一即可，优先JsonPath断言）
      1. 在关键请求中添加响应断言，添加断言数据即可。
      2. 如果是JsonPath断言，则需要填写JsonPath路径和断言数据
   2. 断言的参数化实现
      1. 断言数据可以读取自csv中的数据，与参数化方式相同
5. 接口测试执行
   1. 手工执行测试脚本
   2. 使用ANT执行批量执行脚本

## 什么是关联，在JMeter中如何实现关联：

什么是关联：某请求的请求数据需要用到前面请求的响应数据，需要把前面请求的响应数据关联到后面请求的请求数据中来。

前面响应数据的提取：在JMeter通过在前面请求中添加后置处理器创建正则表达式提取器，正则表达式提取器中包含：引用名称即参数的名称，[可以任意填写]。正则表达式要根据响应来填写，响应在查看结果树中可以获取。在查看结果树中对响应使用RegExp Tester模式查看，获取到要关联的数据及左右边界，将要关联的数据使用(.\*?)来替代。模板一般为$1$，匹配数字一般为1

后面请求数据的关联：把需要关联的请求数据使用${}正则表达式的引用名称即参数替代即可。

## 简述HTTP请求默认值：

如果HTTP请求没有填写，则按HTTP请求默认值进行取值，如果HTTP请求有填写，则按HTTP请求进行取值。一般HTTP请求默认值填写的是协议、服务器域名或IP以及端口

我们做接口测试，可能不止对一套环境进行测试。如果要对多套环境如测试环境、UAT环境、甚至生产环境进行测试，我们则可能需要切换环境，环境的切换只是更改了协议、服务器域名或IP以及端口这些信息。如果设置HTTP请求默认值，那么更改测试环境会非常的方便。

## Json断言如何做

作用：对Json的返回结果进行验证

在哪里添加Json断言：在要断言的请求处添加断言，选择Json断言。

如何填写：JsonPath的填写，$.节点名称.子节点名称。在Expected Value中填写预期结果即可。

## Postman和JMeter的区别：

1.创建接口用例集（没区别）

Postman是Collections，JMeter是线程组，没什么区别。

2.步骤的实现（有区别）

Postman和JMeter都是创建HTTP请求

区别1：Postman请求的请求URL是一个整体，JMeter分成了4个部分（协议、主机、端口、路径）

区别2：Postman可以在请求中直接填写请求头信息，JMeter需要通过添加HTTP请求头管理器添加请求头

区别3：对于Cookie管理，Postman可以对Cookie做管理。但是JMeter只需要添加HTTP Cookie管理器即可完成Cookie的处理，并且是自动处理Cookie信息，显示JMeter的Cookie管理更简单。

关联：Postman和JMeter都能够实现关联

区别1：Postman在pre-request script可以添加前置请求，获取响应数据，比较容易进行JSON结果的处理，很方便的提取JSON数据。JMeter不仅可以处理Json数据（Json提取器），还可以提取其他数据（正则表达式提取器）

区别2：如果响应中有多组数据需要提取，用Postman处理困难一些。在JMeter中用正则表达式提取器可以很方便的提取。

3.数据用例的实现（有区别）

区别1：JMeter比较适合进行数据与操作相分离，而Postman比较适合数据和操作在一起，显然Postman操作更简单，JMeter更便于维护。

区别2：Postman也支持CSV数据文件的导入，但是每次执行时都需要手工加载数据文件，不方便（半自动化）JMeter可以进行完全自动化，特别是引入ant后效果更明显

4.断言的实现（区别不大）

区别1：Postman自带了很多断言函数，可以直接引用即可，操作非常方便。JMeter也自带了断言，操作非常直观。要说区别就是Postman用函数进行断言，JMeter用元件进行断言。

区别2：JMeter支持正则表达式断言，Postman不支持。

区别3：JMeter的断言更丰富，但是Postman需要通过编程来实现同样的效果，此时Postman的断言难度加大了。

5.执行（没区别）

区别1：默认执行，Postman是不能够保存结果的。JMeter可以保存结果，虽然很丑。但是Postman可以通过newman实现批量执行和保存结果，JMeter可以通过ant实现批量执行和保存结果。

6.结论：

Postman比较适合手工接口测试，因为简单，同时它能够实现半自动化。

JMeter比较适合自动化接口测试，因为功能强大并且可以保存脚本，批量执行设置很容易。

Postman一般用来做接口测试，用来发现Bug，验证后台程序

JMeter一般用来做自动化测试，做冒烟测试。

## 请详细阐述接口测试和UI测试在测试活动中是如何协同测试的？

接口测试和UI测试这两块其实是有一部分是重叠的，UI测试是通过前端写的界面，来调用接口，而接口测试是直接调接口。所以排除前端的处理的逻辑和调用的正确性，在理论上接口测试是可以覆盖所有的UI测试的业务逻辑。但实际过程中，如果只是在接口层覆盖所有的业务流，在UI上只测试前端的逻辑，最终的结果可能会是忽视很多原有的功能点，导致了UI测试的不充分。所以存在多人分工且时间充分的时候可以尝试接口去做业务流的全覆盖，否则不要轻易尝试。

## 依赖于第三方数据的接口如何进行测试？

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

## 接口测试中依赖登录状态的接口如何测试？

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

如果是在JMeter下，添加一个HTTP Cookie管理器即可。

## HTTP接口测试和Web Service接口测试区别是什么？

区别是有的。主要是传统ws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。以传统工业标准的ws返回数据为例，返回结果需要包装在一个soap协议指定的语法格式中。即使你只需要简单的返回字符1，也需要包装在协议种返回，协议描述了成功失败否，结果值等。而普通的get，你输出1，在调用端得到字符1。

web service和http接口的区别在于：

1.接口中实现的方法和要求参数一目了然。

2.不用担心大小写问题。

3.不用担心中文 urlencode 问题。

4.代码中不用多次声明认证（账号，密码）参数。

5.传递参数可以为数组，对象等。

## 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。

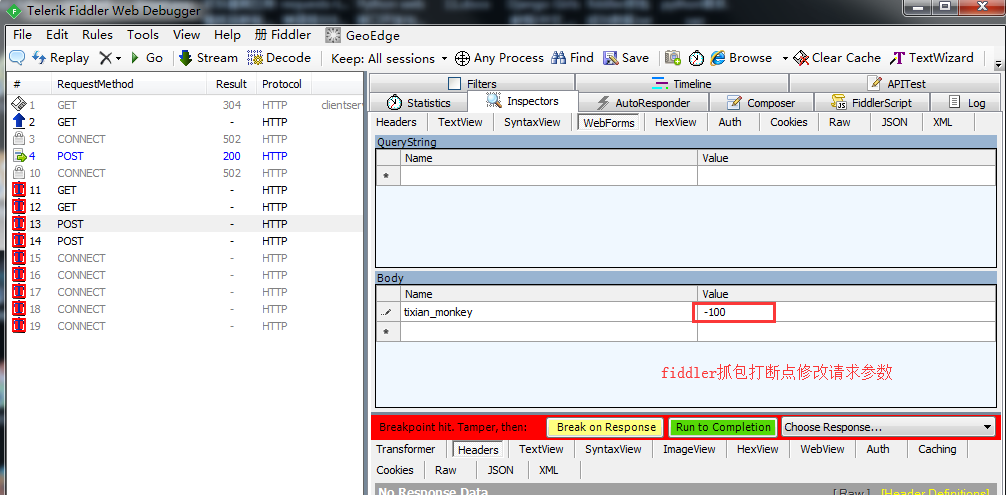
面试官出这个题，主要是想知道你是不是真的做过接口测试，毕竟现在很多小伙伴简历都是写的假的（你要不写估计面试机会都没有，没办法，为了生存，能理解）

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

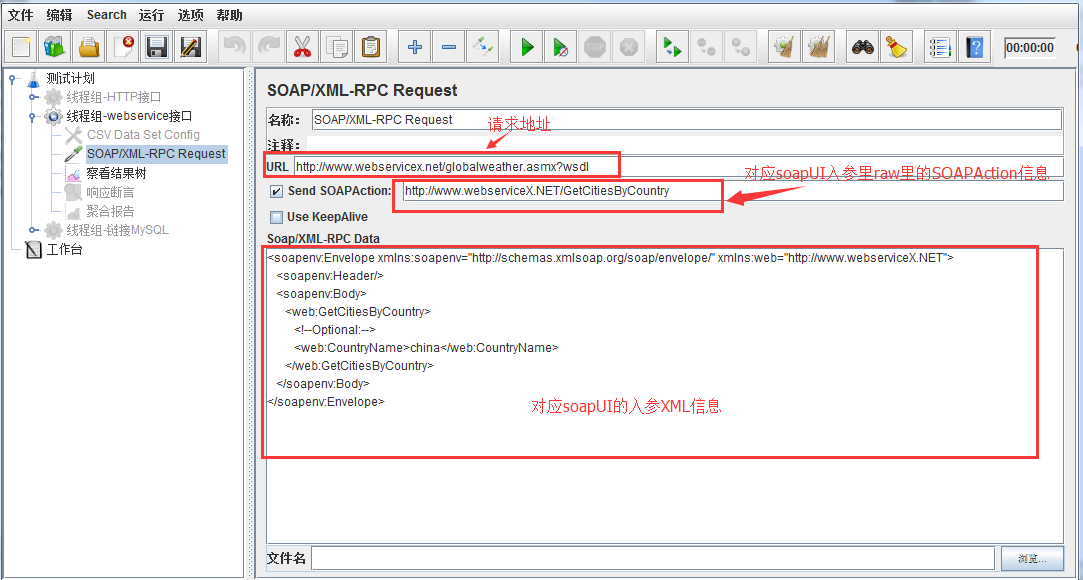
3.检查系统的安全性、稳定性

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）

## webService接口是如何测试的（可不背）



## 没有接口文档，如果做接口测试？

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

3.还有一些接口请求会保存在数据库中，可以找开发要到

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

## 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. 请求报文比较容易出现的是用错了请求方法，URL地址不正确，请求读取的是参数名称而不是参数中的值。也有可能是请求头的Content-Type值不正确，请求Body也可能写错了。
3. 返回报文可以分析出现了什么错误，通过HTTP状态码分析，如果是4xx错误则一般是用户填写请求不正确，5xx错误则是服务器出错了可直接提交Bug让开发跟进。如果状态码是200，则是业务逻辑错误即程序Bug

## 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，我们的业务过程是用户在浏览器输入数据🡪转为请求🡪后台程序（接口）处理🡪产生响应🡪转为结果页面

先抓包看请求报文，对着接口文档，看请求报文有没问题则说明正确。有问题就是前端发的数据不对，属于前端没有把用户数据正确转化为请求。如果前端输入后，无法生成请求，若有Bug则也只能是前端Bug。

请求报文没问题，那就看返回报文，若返回报文正确而前端页面显示不正确，则也是前端解析响应不正确，属于前端Bug。如果返回的数据不对，那就是后端开发的问题咯。而大多数时候是后端Bug更多，并且后端Bug会更严重，影响的是业务逻辑和数据的正确性。

# 性能测试技术专题

## 性能测试的术语

**事务：**一个任务从开始到结束的过程

**响应时间：**一个任务（事务）完成所花费的时间。响应时间S=网络传输时间+Web服务器处理时间+DB处理时间+Waste浪费时间

响应时间的258原则：

S<=2s 优秀 2s<=S<=5s 良好 5s<=S<=8s 一般 S>8s bug

也有123秒之说，含义与上相同。

**吞吐量：**每秒服务器处理的请求个数或事务数

从业务角度看，吞吐量可以用每秒请求数，每秒事务数，每秒页面数，每秒查询数等单位衡量

**并发：**同一时刻不同的用户做同一个操作或对同一个服务器做不同操作（在JMeter体现在线程数）

绝对并发：同一时刻不同的用户做同一个操作

相对并发：同一时刻不同的用户对同一个服务器做不同操作

**并发数：**同时操作的用户个数（在线程组中进行设置）

并发用户数的计算：二八原理（20%的时间内完成80%的交易量）

**场景：**模块用户的一个真实操作过程，场景相当于性能测试的测试用例。

思考时间：实际的场景中，用户操作之间的时间间隔，这个称为思考时间。对应到性能脚本中，就是每个脚本执行的间隔时间，在JMeter中使用定时器实现（固定定时器、高斯随机定时器都可以）

## 断言的种类（响应断言和JsonPath断言）

响应断言的操作主要包括四种：

包含：预期结果应该包在实际结果当中

匹配：使用正则表达式进行匹配

Equals：完全相等（很少用）

Substring：与包含差不多，但不能使用正则表达式匹配

## 正则表达式提取器的配置方法

引用名称：定义一个变量名

正则表达式：提取内容的正则表达式

模板：如果表达式中有多个小括号，$1$表示使用第一个括号的结果

匹配数字：如果提取出来的结果有多个，1表示第一个，0表示随机取值，-1表示取所有结果（是个列表）

## 性能测试指标及通过标准

吞吐量：大于期望值

响应时间：小于5s

错误率：低于0.01%

CPU：低于85%

内存空闲率：高于10%

网络：低于50%

## 做过性能测试吗

我之前做过一些性能测试，使用的是JMeter开展的性能测试，将性能测试的整个过程进行介绍。

## 性能测试的核心过程

1. 确定性能需求和性能测试方案。需要确定性能测试范围（覆盖哪些场景），性能测试策略，并发用户数和加压方式等。
2. 录制性能测试脚本，增加脚本使性能测试脚本尽可能真实接近性能测试的实际情况
3. 收集性能测试结果，分析性能瓶颈（Bug）
4. 提交Bug，出具测试报告

## 建立性能测试脚本的基本步骤

1. 新建线程组（确定线程数和Rand up时间）
2. 添加HTTP取样器，设置HTTP请求数据
3. 根据性能场景进行参数化和关联等操作
4. 添加断言—响应断言
5. 添加监听器-聚合报告或汇总报告

## 性能测试方案的核心内容

性能测试方案是开展性能测试前的核心内容，决定着性能测试如何开展的指标标准。其主要包括性能需求调研过程（并发用户数的计算过程），性能测试模型的建立，性能测试策略（加压方式等），性能测试计划，用例等。

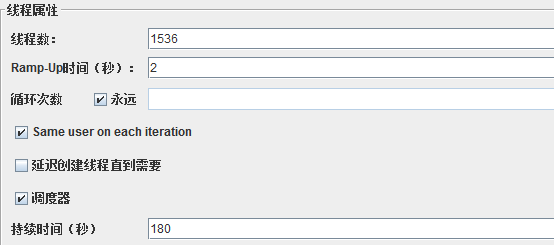
## 并发用户数的计算方法

在性能测试有个核心内容也是在谈性能测试时绕不过去的问题，就是并发用户数的计算。目前做性能测试采用得比较多的并发用户数的评估策略是20-80法则。

通过与项目团队（含客户、产品）评估出系统高峰期和高峰时段，得到该高峰时段的用户数量。假设本电商系统高峰时段为双十一上午10点为高峰期（做秒杀/团购活动）的用户数量是10W用户量，高峰时段为15分钟。那么我们可以通过20-80法则来计算并发用户数，即80%的用户会在20%的时间内完成交易，计算结果为：(80%\*10W)/(15分钟\*60秒\*20%)=80000人/180秒=444.44人

因为公司产品不只是为了满足当前的性能需求，还要能够满足未来3年的性能需求，而公司业务是增长的（不增长会被打死的），预计增长为20%（领导说了算，他开心就好），那么3年之后的并发用户数为：444.44人\*1.2\*1.2\*1.2=768人，则得到每秒并发用户数768人。

接下来，用JMeter录制脚本获取1个用户完成该业务的平均响应时间，如果完成这个业务的时间是2秒，则最少需要768\*2=1536人的虚拟用户才能够完成全部交易。则在JMeter中设置线程数为1536人，Ramp-up值为2，持续时间为15分\*60\*20%，也可以先设置5分钟看效果。



## 负载测试、可靠性测试、可用性测试的定义，有什么区别

负载测试是持续加压的表现，测试系统处理数据的能力

可靠性测试在规定的条件，规定的时间内，软件不引起系统失效的概率

可用性测试对用户界面及功能的是否可用进行测试

## 什么时候做负载和性能测试

我们一旦完成界面（GUI）测试，我们就可以执行负载测试。现代的软件系统架构庞大而复杂的，而单用户测试主要是侧重于系统组件的功能和用户界面测试，应用测试侧重于整个系统的性能和可靠性。例如：一个典型的应用测试场景，描绘了1000个用户，同时登录到系统。这就产生了问题，如系统的响应时间是多少，它会崩溃么；是否兼容不同的应用软件和系统平台；它是否可以支撑成千上万用户等，这时我们就需要做负载和性能测试。

## 你如何识别性能瓶颈

性能瓶颈可以使用监控监测，这些监控器可能是应用服务器监控器，web服务器监控器和网络监控器。它们帮助找出场景中导致响应时间增加有问题的区域，通常所做的是测试指标是性能响应时间，吞吐量，点击率，网络延迟图等。

## 如何发现应用服务的相关问题

1. 通过某些事务的运行，判断是否在应用代码层未进行调优导致事务响应事件过长
2. 通过实时监控工具（nmon等）监控分析
   1. 系统在运行过程中CPU是否稳定运行或CPU耗用是否过高
   2. 在系统运行过程中其内存是否存在内存泄漏现象
   3. 打开应用相应日志，分析在运行过程中是否存在交易报错并获取错误原因查看是否由于代码原因导致交易发生错误。

你做过性能测试吗？/你们公司性能测试流程是怎样的？

1. 拿到性能需求后，根据需求规格和业务主流程进行性能场景分析，性能模型设计。然后进行工作量评估，编写性能测试方案。
2. 方案完成后，搭建性能测试环境，准备性能测试数据，然后开发性能测试脚本。
3. 开发脚本的主要方式为：一种是手工通过Fiddler抓包，然后再导入成JMeter脚本。
4. 主要脚本功能实现后，然后对脚本进行增强，例如：实现参数化，关键以及添加逻辑控制器等。
5. 脚本准备完成后，使用JMeter工具进行性能测试，收集性能数据及监控性能资源。
6. 测试完成后，对性能测试结果进行分析。如果性能结果不符合预期，则配合开发人员进行性能调优，进行回归测试，直到达到预期性能。
7. 性能测试结果通过后，编写性能测试报告。

# App测试常见面试题

## WEB测试和APP测试的区别：

一个系统如果比较成熟，可能会发展成为一个后台，若干个前端，包括：WEB端、app端（IOS、Android）、小程序端、WAP端（比较少）、微信公众号端，会产生多个。因为前端比较多，所以我们测试也就比较复杂，在这里需要区分多种端的测试区别。

WEB测试和App测试从流程上来说，没有区别。都需要经历测试计划方案，用例设计，测试执行，缺陷管理，测试报告等相关活动。从技术上来说，WEB测试和APP测试其测试类型也基本相似，都需要进行功能测试、性能测试、安全性测试、GUI测试、兼容性测试等测试类型。

他们的主要区别在于具体测试的细节和方法有区别，比如：

在功能测试上几乎没有区别，只是APP的测试平台是手机上，一般选择真机（Android和IOS分别测一款），WEB的测试平台是在PC上进行测试。

性能测试，因为WEB端和App端共一个后台，所以后端性能是同一回事。主要体现在前端性能上的区别：WEB测试关注网页页面加载时间，App测试关注App在手机端渲染的时间。一般WEB的前端性能问题很少，App的性能问题相对多一点（我没有遇到过），有些App测试还需要考虑流量测试和耗电量测试。就目前而言，流量测试和耗电量测试都很少做了，因为对于App的流量测试因为流量越来越便宜，网络越来越快的情况下，流量测试在绝大多数公司已经不做了，只有那些视频类App可能会有考虑。耗电量测试也是测试转入后台，对电量的损耗情况，只要不是频繁读取后台数据和进行联网设置，耗电都不大，其实测试的意义也很小。

兼容性测试：在WEB端是兼容浏览器，在App端兼容的是手机设备。WEB因为是测试兼容浏览器，所以需要使用不同的浏览器进行兼容性测试（常见的是兼容IE6，IE8，chrome，firefox）。如果是手机端，那么就需要兼容不同品牌，不同分辨率，不同android版本甚至不同操作系统的兼容。（常见的兼容方式是兼容市场占用率前N位的手机即可）。因为浏览器的自适应性，在WEB兼容性测试时一般只做IE和Chrome的两款浏览器的测试。而App兼容性也开始越做越好，我们也只会对IOS的最新款和Android的最新款进行兼容性测试即可，如果有更高要求，会使用Testin这样的商业工具也可以做兼容性测试。

安装测试：WEB测试基本上没有客户端层面的安装测试，但是App测试是存在客户端层面的安装测试，那么就具备相关的测试点。安装在App中对自己所测设备进行一次安装与卸载测试即可（因为没有太多可选项），如果要做安装与卸载的兼容性测试，可在Testin平台下进行测试。Testin兼容性测试的原理：针对于所选机型进行一次安装与卸载测试，并进行一定时间的Monkey Testing。

还有一些App测试所独有的，例如： 交叉事件测试，操作类型测试，网络测试（弱网测试，网络切换），版本升级测试等

交叉事件测试：就是在操作某个软件的时候，来电话、来短信，电量不足提示等外部事件。会将App程序转入后台，我们要验证转入后台与切换到前端的操作上是否正常。

操作类型测试：是指手势测试和横屏测试。手势测试是指滑动、单指、双指操作的行为是否满足预期。其实这是属于操作系统存在的东西，根本不需要测试。一般只做横屏测试，验证该App能够横屏并在横屏下是显示正常的即可。

网络测试：包含正常网络、弱网、无网和网络切换测试。正常网络测试在功能测试同时已完成，无网测试只需要测试登录页面或者是固定首页，检查其数据和展示是否正常即可。弱网测试需要测试弱网所造成的用户体验（即在弱网下提交失败的操作结果检查），重点要考虑回退和刷新是否会造成二次提交，一般在关键交易上。App的弱网测试可通过Fiddler来完成，模拟弱网可在Fiddler🡪规则🡪Performance🡪Simulate Modem Speeds实现。在弱网下会导致的是用户提交失败以及二次提交，二次提交只针对于关键交易类场景进行测试即可，二次提交可使用接口工具重复提交某个请求即可。

从系统架构的层面，WEB测试只要更新了服务器端，客户端就会同步会更新。而且客户端是可以保证每一个用户的客户端完全一致的。但是APP端是不能够保证完全一致的，除非用户更新客户端。如果是APP下修改了服务器端，意味着客户端用户所使用的核心版本都需要进行回归测试一遍。

还有升级测试：升级测试的提醒机制，升级取消是否会影响原有功能的使用，升级后用户数据是否被清除了。

## App出现crash原因有哪些？

为什么App会出现崩溃呢？百度了一下，查到和App崩溃相关的几个因素：内存管理错误，程序逻辑错误，设备兼容，网络因素等，如下：

1.内存管理错误​：可能是可用内存过低，app所需的内存超过设备的限制，app跑不起来导致App crash。

或是内存泄露，程序运行的时间越长，所占用的内存越大，最终用尽全部内存，导致整个系统崩溃。

亦或非授权的内存位置的使用也可能会导致App crash。

2.程序逻辑错误：​数组越界、堆栈溢出、并发操作、逻辑错误。

e.g. app新添加一个未经测试的新功能，调用了一个已释放的指针，运行的时候就会crash。

3.​设备兼容：由于设备多样性，app在不同的设备上可能会有不同的表现。

​4.网络因素：可能是网速欠佳，无法达到app所需的快速响应时间，导致app crash。或者是不同网络的切换也可能会影响app的稳定性。

## app对于不稳定偶然出现anr和crash时候你是怎么处理的？

app偶然出现anr和crash是比较头疼的问题，由于偶然出现无法复现步骤，这也是一个测试人员必备的技能，需要抓日志。当出现偶然的crash时候，这时候可以把手机拉到你们app开发那，手机连上他的开发代码的环境，有ddms会抓日志，这时候出现crash就会记录下来日志。

也可以自己开着logcat，保存日志到电脑本地adb logcat \*:E >d:\hello.txt

# 微信小程序测试

## 小程序的特点

**小程序的介绍与认识**

是一种微信开发不需要下载安装即可使用的应用。

**小程序的架构**：包含View视图层、AppService逻辑层。View层用来渲染页面结构，AppService层用来逻辑处理、数据请求、接口调用，它们在两个线程里运行。视图层使用WebView渲染，逻辑层使用JSCore运行。

**小程序的限制：**

1）数量限制：小程序一个应用只能同时打开5个页面；若忽略5个页面的限制，这时如果跳转逻辑超过5个页面会出现跳转打不开的情况；若已做到跳出5个页面的限制，需要特别查看各个页面间的跳转情况，重复进入的页面以及关闭页面能否返回上一层页面。

2）大小限制：小程序源码打包后的大小限制为1M，因此源码中的图片和icon和数据等都需要压缩。这里就可能出现数据和图片丢失，不清晰等问题

**小程序的环境类型**

由小程序的版本类型可以看出，我们一般会准备三套环境：

1、发版访问测试环境

2、体验版访问预发布环境

3、正式版访问生产环境

## 小程序和Web/App有什么区别

**小程序设计理念：**微信小程序开发相当于重新做了一个不同于传统形式的APP，从开发、设计、测试、升级都是单独的一套。微信小程序和原生APP以及轻应用都不同，它更像是微信功能的延伸，更加注重场景化。

**小程序本质内容：**但从本质上讲，它依旧没有摆脱原生APP开发以及轻应用所面临的两个问题：基于H5的特性和功能上的限制，微信小程序开发虽然无需安装可以即开即用，能够很大地减少手机的存储空间占用，但是对于网络的依赖也更加严重，轻量的定位在功能体验上也会不如传统APP开发出色。

## 小程序常见测试点：

**1.功能测试**

按功能模块测试、按业务流程测试、按数据流向测试、同一功能不同的入口有效性的检查、交互性检查等

**2.兼容性测试（微信在适配上已经做得比较好了）**

对于已上线的小程序，有可能会因为**微信版本**升级之后导致对部分小程序的组件支持产生冲突，手机端微信上查看的小程序页面出现样式有异常，普通的手机APP会有屏幕兼容性的问题，小程序同样有这样的问题，只不过相对少了些。

**3.接口测试**

小程序的后台接口跟其他的客户端后台接口测试类似，直接按照常规的后台测试来开展就可以。

**4.登录方式**

小程序主要有两种登录方式，一种是使用微信授权登录，另一种是手机号登录。在小程序中，没有注册功能，一般只有首次登录这种场景。

**5.交易**

小程序支持交易，那么它与微信的钱包、卡包都是可以交互的。如果有交易功能，需要验证各种交易场景。微信小程序不支持支付宝支付。

**6.性能测试**

这里的性能测试考虑的是客户端的性能，服务器的性能则按照传统的服务器性能测试方案即可。小程序的客户端性能和网页的性能测试非常类似，性能的常用指标也大致相同。包括页面的白屏时间，首屏时间，资源占用，页面渲染时间，帧率等等。

# adb 命令

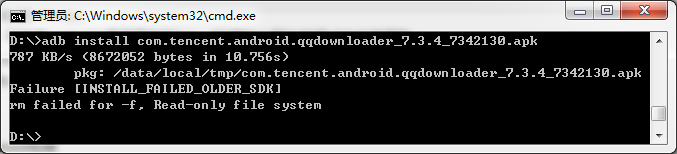
adb是运行在 PC 端的程序: 可以通过它对 Android 应用进行安装、卸载及调试。adb的常用命令有：

1. **adb devices** , 获取设备列表及设备状态



1. **adb install xx.apk用于安装**

安装成功，返回成功提示 "Success"；



1. **adb shell pm list package**

Package Manager , 可以用获取到一些安装在 Android 设备上得应用信息

-s：列出系统应用

-f：列出应用包名及对应的apk名及存放位置

-3：列出第3方应用的apk名，也就是用户安装的应用程序名称（常用）

1. **adb uninstall 应用名称用于卸载**

adb uninstall 后面带的是应用的包名，而不是应用名。

1. **adb logcat**

有时候我们在手机程序上的日志要在其他地方调试，然后要看里面的Log日志。在cmd窗口中输入如下命令：

|  |
| --- |
| //格式1：需要打印时间和级别是Error的信息  adb logcat -v time \*:E  //格式2：将日志保存到电脑固定的位置，比如D:\log.txt  adb logcat -v time >D:\log.txt |

通过查看日志信息，定位当前操作是否为Bug，是什么样的Bug。在咱们做App测试时，当出现carsh和anr时，会实时跟进日志信息。（日志开发看得懂，直接保存日志到禅道附件即可）

# adb monkey

adb monkey是一款稳定性测试工具，通过随机点击屏幕一段时间，app不会出现崩溃，无响应，能够正常维持运行的测试。是由android系统自带应用，可以通过adb shell monkey直接使用。

1. **monkey 命令执行**

|  |
| --- |
| adb shell monkey -p com.ren.che --throttle 500 -s 100 --ignore-crashes --ignore-timeouts --ignore-security-exceptions --ignore-native-crashes --monitor-native-crashes -v -v -v 1000>D:\android\monkey\_log\16v3.txt |

adb 的详细用法和说明，请参考

<https://www.cnblogs.com/laoluoits/p/12750883.html>