**就业课程总结**

**一、面试常问（开始）**

**1.自我介绍**

在毕业后一直在从事软件测试方面的工作。在做测试期间，主要测过web端和App端相关的项目。在上家公司的话我的主要工作职责是：需求分析与确定、功能测试、自动化测试、接口测试。

在技能上的话，我掌握了功能测试、接口测试、接口自动化测试、UI自动化。jmeter、fiddler、badboy等常用测试工具都能使用，以上就是我的自我介绍，您还有什么想问的吗。

**2.你们公司的项目流程**

首先接到一个新的需求，产品经理召开需求评审会议 介绍功能细节，需求会议后，我们老大会根据需求制定测试计划。我们测试人员按照计划，进行用例设计编写，用例编写完成后进行用例评审。评审结束，我们就把用例更新并上传到用例管理工具。

接下来等开发提测，提测后，我们测试介入测试。

首先进行冒烟测试。

冒烟通过后，接着进行SIT测试（一天执行80条用例），有bug就指派给相应的开发，等开发修复后指派回来给我们测试，我们进行验证确认bug修复。

根据情况接下来可能还有两到三轮SIT,每轮抽取一定比例的用例测试并输出小结。

接着进行回归测试。

最后输出测试报告，让产品来验收，上线。

**二、项目**

**1.讲一下你最近做的项目（金融）**

我最近做的是一个金融项目叫金开贷，是一个资金管理平台,主要是为投资人或有借款需求的用户提供的平台。

我们项目由前后台组成，前台就包括：首页、我要投资、我要借款、信息披露、个人中心。

后台是一些管理模块，例如：投资管理、会员管理、标的管理、系统配置以及一些推广模块等等。

我们平台的核心业务主要是投资和借款

那我就拿我要借款这个业务来说，我们分为抵押贷和信用贷。

那我说一下信贷的业务流程吧。

首先一个新用户先自己注册一个账号并登录，填写并提交相关信息，开通和我们合作的银行存管账户。用户提交信息后,生成一个授信订单，风控小组收到授信订单，并调用个人征信系统（百行征信）查询用户的个人征信并返回结果，风控会对结果进行审核，审核通过后，风控将结果回调给信贷，给用户授信。

接下来用户进行提现，根据用户选择的提现金额，提现方式、还款方式等信息，生成一个提现订单，信贷会把该订单送到风控进行提现审核，根据审核结果，会有三种状态，01状态就是审核通过，02状态需要人工审批（大金额提现），03状态就是审核不通过(提现大于授信)。假如回调01状态，提现通过后，信贷会将提现订单推到资产方，资产方生成一个还款计划，并放款给用户。放款后，放款的结果会返回给信贷，这就是信贷的一个大致流程。

**2.给你一个模块，你如何展开测试工作（让你负责一个模块，你怎么测的）？**

首先分析一下需求，提炼出业务和功能点，然后从不同角度去进行测试。

我会对功能进行验证。业务流程、功能点、正常的场景、异常的场景都去测试到位，业务逻辑也去进行测试.

然后看UI是不是和高保真图一致，测试元素是否齐全、分布合不合理，有没有错别字

兼容性的话，如果是web端，那就需要考虑电脑的系统、不同的浏览器、不同的浏览器版本

如果是移动端，那还分为安卓和苹果，安卓要测试不同机型，不同的系统版本，屏幕是否兼容

然后是易用性测试，看看模块是否使用方便，有没有相关的提示。

还要进行性能测试。

**3.你们多久上一次线（多久发一次版本）**

这个要看需求的大小，也要看看客户的要求，有可能一周发一次版，一个月上一次线的也有

**4.你们项目组有多少人，人员比例？**

我们项目组有3个测试14个开发。公司的话有50多人

**5.你负责哪块，你跟你同事怎么分配工作的？**

我之前做的项目里有web端和app端，我主要负责web端的测试

**6.你做功能和自动化测试比例是多少？（你花多少时间做功能，多少时间做自动化）**

这个没有绝对的，如果现在这个版本的优化和新功能跟我负责的相关，那我近期肯定就是就做功能偏多

如果大多跟我无关，我近期会去做自动化测试更多

**三、需求相关**

**1.你们公司有没有去需求评审环节？讲一下你们公司需求评审流程**

有。首先会有邮件通知，邮件内容大致就是参与需求评审的人员，会议时间和地点，对什么需求进行评审。

在会议上我们的产品人员会对需求进行讲解，一般会讲业务、功能场景、功能点等。我们有问题可以现场提出，然后进行确认解决，解决不了的就先记录下来，在后续解决。

然后我们的会议记录人员会把此处评审形成的文档用邮件发给与会人员。

**2.你们公司需求会议有哪些人参加？**

参与会议的人有开发团队、测试团队、产品。有时除了我们项目组的人，还有一些其他人员，一般都是客户或其他相关平台的负责人。

**3.给你一份需求你如何分析**

拿到一份需求，我会先通读一遍，全面分析一下需求。第二遍，我会去分析需求的可行性，看看需求描述合不合理。接着，我会借助xmind思维导图工具提炼业务，按轻重缓急来提炼，提炼出主流程、分支流程、异常流程等。我还会去提炼隐性需求，记录下有疑问的地方，找产品进行确认。

一个大的需求，还要分析需求的发布时间（就是需求做这么久合不合理）合不合理，考虑需求中的人员配备是否合理、软硬件配备是否齐全

**4.让你负责一个需求（复杂需求、紧急的需求），你如何安排工作**

首先是需求阶段，我会对当前这份需求分析透彻，并及时澄清，完成评审。然后编写出相应的测试计划，对计划进行评审；

接着是设计阶段，我会从需求中提炼出业务场景和功能点，并采用合适的用例设计方法结合场景设计用例，保证用例的覆盖率。再通过用例评审加强用例的健壮性；

然后是执行阶段，我会严格按照测试要求去测试。每天都反馈一下测试进度，有多少bug，什么原因导致的，什么时候去解决；

最后收尾阶段，输出测试小节、测试报告，然后去完成线上环境验证的工作。

**5.当一个项目没有需求，你怎么展开测试工作**

1.尽可能找相关的参考文档。

2.及时与相关的负责人进行沟通，比如领导、团队成员。

3.寻找类似的产品进行参考，借鉴有价值的部分。

4.根据自己的以往测试相关产品经验和判断，输出初始可参考方案并提交给相关人进行讨论和评审。

**6.需求变动频繁，你如何应对**

在需求变化频繁的情况下，作为测试人员，应该搞清楚以下几点

 1.哪些需求发生了变化

    2.这些需求变化后，对测试[工作](javascript:;)会产生哪些影响。包括会不会影响[测试用例](javascript:;)，如果影响，会对哪些用例产生影响。当发生较大改动时，还要明确是不是影响到了测试方案，甚至是测试计划？

    3.明确这些变化，会对自己的工作进度产生多大的影响。当发现自己的大部分用例都受到影响，需要修改时，应该第一时间向上级反映情况。由他定夺解决方案，而不是自作主张，或是一声不响。

    关于如何确定哪些需求发生了变化，最好的工具当然是需求跟踪矩阵了。需求跟踪矩阵是从开发和测试两条线，同时进行跟踪的。但是，要实现需求跟踪矩阵，可不是容易的事情。尤其对于需求变化频繁的公司来说，基本可以断定他们是没有配置管理的。那么需求跟踪矩阵就根本没有实现的可能。但是，公司没有，不代表我们自己没有。

    公司的测试任务分配，一般都是按照子系统或者是模块来分的。比如TesterA测试子系统A，TesterB测试子系统B。那么这时，我们完全可以只针对自己测试的子系统，完成需求跟踪矩阵。我们只需要自己维护测试线的需求跟踪，至于开发线，那可是管不着了。但是这里还是有点问题。我们一般理解的需求跟踪矩阵，是将SRS分成子SRS，然后针对每个子SRS，建立跟踪关系。但是忽略了各个子SRS之间的关联关系。要想把各个子SRS划分得没有一点联系，那是不太可能得。如果有两个子SRS，称为SRS1和SRS2，你测试的是SRS1相关模块，而SRS2相关模块是别人测的。当SRS1相关开发人员告诉你需求已变更后，你分析后得知影响到了SRS2，那么你必须和测试SRS2相关模块的测试人员及时沟通。你能保证做到这点，但是对方未必会在SRS2发生变化时，及时通知你SRS1受影响的部分。这时你需要事先和测试组的[**其他**](javascript:;)人员充分沟通，让他们及时通知可能会影响到你的变更。如果他们不及时通知，你就完全有推卸责任的理由了。对于开发人员也是一样，他们擅自改了程序，不通知变更SRS，或者不及时通知变更，你也完全有理由推卸责任了。

    我们的目标肯定是要把测试工作做好，而不是自己没有责任就好了。当建立了自己的需求跟踪矩阵以后，就可以快速定位变更部分，而且目前企业没有配置管理的理念，所以可以及时变更你的用例，方案，甚至是计划。当发现受变更影响的部分非常多时，应该及时通知上级，让他们了解情况，并做出决策。

    总结一下的：需求变化快，测试工作需要重新写用例，方案，甚至计划。这时遇到问题：

    1.得不到及时通知

      解决方案：没有办法，只能事先和开发人员以及测试组内部人员事前声明，不及时通知的部分，一概不付责任

    2.不知道需求变化后，对哪些工作产生了影响

      解决方案：公司没有配置管理，没有需求跟踪，那么就建立自己的小的需求跟踪。只跟踪测试线。把自己的工作做好，并及时通知其他受影响的测试人员。

    3.发现自己大部分用例，方案，甚至计划要重新修改，而没有充足时间

     解决方案：及时通知上级，让他给解决方案，而不是自己苦干。

**四、测试计划相关**

**1.写过测试计划吗？测试计划一般什么写？**

写过。测试计划一般是在需求评审之后进行编写

**2.你们公司测试计划包括什么内容？**

我们公司的测试计划包括：简介、测试约束、测试环境与资源、测试任务进度安排，风险评估，测试交付文档（测试计划、xmind图、用例、第一轮测试小结、测试报告、BUG数据）

**3.如何写好一份测试计划？**

首先从内容上来说，测试计划应该包括：简介、测试约束、测试环境与资源，测试任务进度安排，风险评估，测试交付文档

第二，一份测试计划应该着重于写好测试任务进度安排。测试任务进度是我们项目测试的时间轴，规定了各阶段开始和结束时间，对应负责人是谁。

第三，一份好的测试计划还需要经过评审。

**4.现在有个版本，十五天，你要怎么安排**

先问问面试官，十五天是工作日还是自然日。

再就是要确认十五天里是否包含需求

假设包含

在我们项目流程里，最花时间的有三个阶段：需求、设计、执行阶段。那么除去需求阶段，还有10-11天的时间再从里面拿出1-2天进行测试准备和收尾等，那么还剩下十天，一般就是安排设计用例4天，执行用例六天

**5.测试方案和测试计划的区别？**

测试方案内容一般有：概述、测试资源、测试环境、测试策略

和测试计划的区别就是，测试计划着重于做什么，而测试方案侧重于做什么

**五、用例**

**1.有没有写过测试用例？常见的用例设计方法有哪些？**

写过。常见的用例设计方法有：场景法、等价类、边界值、判定表

所谓等价类，指的就是在某个输入域集合中，每一个输入条件都是等效的

等价类一般分有效等价类和无效等价类，有效等价类就是符合需求的，输入合理的数据

无效等价类就是不符合需求的，不合理的数据

场景法，就是通过运用场景来对系统的功能点或业务流程进行描述，以提高测试效果的一种方法

比如说用户借款的场景

基本流：登录，点击借款，生成订单

备选流：未登录，账号密码错误，风控审核不通过，用户退出系统

然后根据基本流和备选流来生成不同的场景

然后根据这些场景编写测试用例

边界值，就是一种对输入或输出的边界值进行测试的一种黑盒测试方法。

举例：出借最大金额，最小金额

判定表，是分析和表达多逻辑条件下，执行不同操作的情况的工具

例如：用户填写个人数据时

**2.你们公司的用例设计模板？用什么工具写的？**

用excel表格编写的

我们公司的用例模板：用例编号、用例标题、需求id、前置条件、操作步骤、预期结果、用例类型、优先级、创建人

**3.如何保证用例覆盖率？\如何保证用例质量？**

第一，对需求要分析透彻，要澄清和解决 需求和设计文档中的问题。

第二，要根据需求罗列出功能点，用思维导图来提炼测试点，还要进行测试点评测

第三，根据提炼的测试点编写用例，根据不同的功能和场景，采用合适的用例设计方法。用例的排版要规范，要一个一个模块，一个一个点去写

第四，要进行用例评审、对用例查漏补缺

第五，必须严格按照用例去执行。执行时，发现有问题的用例、无效用例或是漏写用例，要及时维护更新用例。

**4.你们项目写了多少用例？**

我不知道，这个我还真没去统计过。

我自己近期写过多少用例我知道，小版本一般就两三百条，版本时间长一些就差不多有一千条用例，我记得我最少的一次就写了几十条用例。

**五、缺陷相关**

**1.你们公司的bug流程**

一条用例执行失败，就转bug，定位到BUG并上传到缺陷管理工具禅道，填写bug相关信息。

接着将bug指派给对应的开发，完成bug修复工作

开发修复bug，指派回对应测试，测试进行验证，完成回归测试，确认修复后，关闭bug。如果测试不通过，继续将bug给回开发

**2.你们公司提bug包括哪些信息**

bug级别、bug标题、所属模块、所属版本、bug重现步骤、bug优先级、bug截图、错误日志、偶现\必现

**3.说一说你印象深刻的bug，如何去分析它的（背）**

1.我印象最深的一个bug就是,我们平台在登录之后，会默认跳转到个人中心，按照流程，会弹出一个温馨提示框，页面其他地方置灰，我当时在登录之后，页面置灰了，但是提示框没有弹出，整个页面卡死。

我通过控制台的网络标签，发现错误报了404，就是请求路径找不到的问题，这种情况就有可能是前端和后台开发人员改了接口地址，这种情况也可以提给前端，也可以提给后端，后来我去和开发沟通了一下，是前端那边的问题，后续解决。

学长的：

1) 我印象最深的是我在做洋葱先生这个项目中，在进行注册账号时。对界面进行多个输入判断的时候，按界面从上到下的顺序--满足条件之后，注册是没有问题的。但是，改变一下输入次序，效验失败。

解决办法  开发重写判断顺序/逻辑

2、我印象最深的就是我在做小花金融这个项目中，出现的bug，一开始在提交贷款金融数字参数时，后台没有反应，界面也没反应，然后我用抓包工具，重新输入贷款参数，抓取url，参数内容，还有响应结果进行对比，发现参数内容是一致，则判断为可能是后台bug，然后查看数据库数据有没有输入进去，然后通过和开发人员的努力，最终解决了

3、做宜农贷项目的时候，在进行注册账号时，输入完账号和密码，在点击接收验证码信息时，手机上收不到验证码信息，从而无法进行下一步注册，而后跟对应的开发反应，定位bug/后台接收不到前台数据请求

4、财务后台系统查询时，查看订单区间大于一年时间会显示查询超时，比如查看2019年1月～2020年2月的所有借款订单，会显示超时。bug原因是查询数据量太大，表关联过多，解决方案：开发定时将数据统计到一张表，之后查询。

5、我上一个项目是一个p2p网络借贷平台的项目，给我印象比较深的bug有两个，第一个就是在平台注册时候使用手机号码注册，但是发现131开头的手机号码不能注册，显示号码格式不正确，还有一个是在登录了平台之后当有跳转界面即会生成另一个窗口的时候，在其中一个窗口点击退出登录，原本进其他窗口的时候都会显示请先登录，但是发现其他的窗口还是处于登录状态，这样的话降低了客户账号的安全性。

解决情况：

分情况：接口进行注册如果可以，那就是前台js校验问题，没有匹配到13开头的手机号，如果前台和接口都不行，就要从前台js和后台代码这块处理

6、场景还原：输入登录用户名和密码，点击登录，无任何反映（没有任何提升和跳转）

bug分析思路：从页面交互看，输入用户名和密码点击登录要么登入成功进入系统，要么就是跳转到系统其它页面，其次就是登录失败给提示，而现在我们没有任何反应

分析有可能是以下几种情况：

情况一：通过控制台（浏览器F12），如果发现控制台并没有JS错误，但是发现有向后台发起一个请求，此时还不能确定是前端还是后台的问题，需要进一步分析，我们可以通过控制台的网络标签，如果这个时候你发现前端登录脚本执行报错导致前端没有对后台登录接口发起调用，那么这个bug则属于前端bug

情况二：通过控制台（浏览器F12），如果发现控制台并没有JS错误，但是发现有向后台发起一个请求，此时还不能确定是前端还是后台的问题，需要进一步分析，我们可以通过控制台的网络标签，如果发现错误404 not fount请求路径找不到的问题，这种情况就有可能是前端和后台开发人员改了接口地址，这种情况也可以提给前端，也可以提供后端，如何是500则说明是后端服务异常，这种一搬有代码执行异常导致的，bug交给后端处理

**4.开发不认可你提的bug，你怎么做（这里想问的就是你的问题推动能力）**

首先不管开发认不认可，我会先把bug记录下来。然后我会去确认一下bug是不是因为权限、环境、网络等因素导致的。

然后我会再去细化分析分析需求，并再次验证bug。接着去和开发沟通，听听开发怎么说。按照开发说的，我会结合需求跟他讲讲，拉上我们的产品一起对这个问题分析分析。

如果确实是bug，我会重新激活bug，督促开发解决bug

如果和开发说的一样，不是一个bug，我会致歉，尊重开发的意见，关闭bug。

**5.产品即将上线还有bug没有解决，你会怎么处理？**

首先我会把没有解决的bug整理一下（什么bug，bug等级、优先级），然后尽可能在上线前督促开发修复bug

如果还有没解决的，我会把整理的文档抄送给相关负责人。我们测试人员没有权利决定能不能上线，这个只能得到了上级明确的回复才能进行上线或者处理问题

**6.你在公司如何定位一个bug**

首先要确认这个bug是前端bug还是后端的bug

我一般会通过抓包去进行定位，看看接口URL是否正确，看看入参是否正确，如果我输入正确，抓包的结果是错的，那可能就是前端的问题。

反之，如果传参没有问题，响应的内容有问题，那可能就是后台的bug

定位完成后，如果是前端问题，我会去按F12在console（控制台）中查看报错信息

如果是后端问题，那么一般就是查看错误日志和数据库来分析定位

**7.你们项目的技术架构是怎么样的**

我们项目的前端是通过HTML、CSS/JS语言来设计实现的，后端呢是用java、PHP、JSP来实现的

然后数据层我们公司是用的mysql数据库

**8.遇到不可复现的bug怎么处理（偶现）**

一定要提交bug，及时截图，记录bug信息，详细描述出操作步骤，尽力去查找出错原因复现bug。记录下来有这么个问题，以后遇到了可能就会了解发生的原因。

然后我会定时去进行测试，每天上午，下午，不定时的去复现，如果问题再次出现可以直接叫开发来看看

比起测试，开发对代码熟悉的多，可能看看，开发就知道问题所在了

**拓展（数据传输）**

1、数据是怎么发送到后端？

2、后端是怎么接收到前端发送过来的数据？

3、后端怎么对前端发送来的数据进行处理？

4、处理完之后又怎么写入数据库，以及给前端返回处理结果？

具体代码怎么处理我们不做重点，关键明白这个理

首先我们的数据处理大概分三层来看：表示层（显示层），业务层、数据层

第一层，表示层，这部分可以用HTML代码，CSS/Javascript代码来实现等。通过前端代码可以实现网页的布局和设计。这层又可以称为显示层。也就是你用浏览器打开能看到的网页。

第二层，是业务层，这层是负责处理数据的。常用的代码语言有PHP,JSP，Java等。通过这些后台处理语言的算法来处理前台传回的数据。必要的时候进行操作数据库，然后把结果返回给前端网页。

第三层，是数据层，这个就是数据库，用来存储数据的。通过业务层的操作可以实现增删改数据库的操作。

具体处理流程如下：

首先你接触到的是这个网页是属于表示层，这个网页一般由HTML标签结合CSS/JAVASCRIPT来实现的。 这时候你要先填入数据。

然后你从前台提交数据（传参）在公司可以通过form表单形式处理你要传参，通过触发一些按钮事件提交到后台，或者也可以用ajax和json格式（也是通过get或者post方式提交）传参，

后台接收后触发相应的后台处理机制，这时候数据会传到后台的代码进行处理。这部分代码根据不同网站可以使PHP,JSP，JAVA等语言处理。 代码根据程序员预设的算法将收到的数据进行处理之后会相应的对数据库进行操作，存储数据等。

最后你成功操作完数据库之后，业务层的代码会再向表示层也就是显示器端传回一个指令通知你表格填写成功

**六、测试报告、上线相关**

**1.测试报告是你写的吗？测试报告的内容有什么？**

不是，测试报告是我们老大来写的。我们只是提交测试的数据和测试小节，包括执行了多少用例、发现多少bug、bug产生原因，还有多少用例没有执行等。

测试报告包括：

引言（目的、背景），测试概要（用例、用例执行相关），测试环境，测试结果bug情况，测试结论，遗留bug的影响。

**2．有没有参与过上线？**

参与过。一般我们上线都会选择在晚上上线，白天使用的人群比较多，晚上的话上线用户相对少一些，给平台带来的影响也会小一些。

**3.上线前，你在做什么？**

上线前，我会把该输出的文档d都进行输出，包括：测试小节报告、用例执行情况、bug清单、未解决的bug情况。发邮件通知相关人员准备上线

**4.上线时，你在做什么？（验收测试）**

上线时，我会配合相关部门进行上线，完成自己负责的模块在生产环境的验证，并及时记录下线上的数据，遇到了问题，要及时记录，督促并协助开发修复bug。

**5.上线后，你在做什么？**

我会输出此次上线的总结报告，可以为后续做参考

然后要留意一下线上环境的运行情况。

上线遗下来的bug要记录下来，在后续解决

**七、环境、linux、数据库**

**1.你们测试用的测试环境是谁给的？linux怎么搭建测试环境？**

我入职的时候公司已经有测试环境了，我自己也会搭建测试环境，我自己在私下有试过搭建

流程大概是：

1 通过winscp/lrzsz中的rz上传tomcat，MySQL安装包，JDK（Java开发环境工具包）到linux下

2 利用tar -zxvf解压缩包命令对jdk,tomcat，mysql进行解包、安装，再配置jdk环境变量。

3 把war包（web程序）放到tomcate指定目录webapps下，再启动服务器即可。（输入startup.sh的路径，直接回车即可运行）

**2.熟悉linux吗?讲一下用过哪些指令，或者面试官给你个场景，让你讲用什么指令去做**

自己讲：要注意文件操作相关的，查询内容，打印部分内容。

实时刷新日志 tail -f

查进程的

查看资源的：top df free

单独查cpu iscpu

和端口相关的 lsof、netstat

打包命令，打zip、tar、gz包

修改权限的命令

遇到不会的就说在公司中没用过，在工作中没用过

vim编辑命令（三种模式）三种模式的区别：命令模式看不到输入的命令，末行命令模式可以看到，编辑模式就是输入的

vim模式中查找关键信息的指令:/ 字符串

**3.日志**

刷新日志 tail  –f  catalina.out

**日志级别：**

info：仅是一些基本的讯息说明而已；

notice：比 info 还需要被注意到的一些信息内容；

warning 或 warn：警示讯息，可能有问题，但是不至于影响到某个daemon 运作

rr 或 error ：一些重大的错误讯息，这就要去找原因了

rit：比 error 还要严重的错误信息，crit 是临界点 (critical) 的缩写，已经很严重了

alert：警告警告，已经很有问题的等级，比 crit 还要严重

merg 或 panic：疼痛等级，意指系统已经几乎要当机的状态

**4.DDL和DML的区别**

DDL是数据库定义语言，例如 create、drop、show

DML是数据库操作语言，例如insert into、delete from、update 表 set id=1 where sex=1、select \* from

**5.常用的聚合函数**

Count、sum、min、max、avg、distinct

**6．分页查询怎么做的**

用limit

**7.多表连接的方法有哪些，他们的区别**

基本连接、内连接、左连接、右连接

基本:select \* from a,b where a.id=b.sid

内连接:select \* from a inner join b on a.id=b.id

左连接：select \* from a left join b on a.id=b.id

右连接: select \* from a right join b on a.id=b.id

**8.视图索引的作用和创建方式（冷门）**

**视图：建立一个虚拟的表，主要用于保护敏感信息，类似软连接，改视图会影响本表**

create view 视图名 as (select 字段名 from 表)

**普通索引：最基本的索引，没有限制**

create index 索引名 on 表面(字段)；

**唯一索引：索引列的值必须唯一，但允许有空值**

create unique index 索引名 on 表名(字段)；

**删除索引**

alter table xx drop index 索引名

**查看索引**

show index from 表名；

**9.存储过程**

用这个例子来讲

DROP PROCEDURE if EXISTS yy; (这个不用说，调试用的)

CREATE PROCEDURE yy(n int)

begin

DECLARE y varchar(20) DEFAULT '';

DECLARE i intDEFAULT 1;

while n>i DO

set y=CONCAT('0000',i);

set y=CONCAT('太平洋-深圳-',LPAD((i),5,0));

insert into fp(id,user,num,sid)values(y,'小陈',18898759532,430223111111111111);

set i=i+1;

end while;

select \* from fp;

END

call yy(20)

**mysql熟悉吗？有没有写过存储过程**

熟悉，增删改查、单多表查询、多表拼接都能掌握。

存储过程写过，之前有一次我们老大有让我造10万条用户数据。

我记得当时那个订单数据包扩订单标号、用户名、手机号、用户身份证号这几个字段

存储过程呢，首先是用create procedure创建存储，开始是begin，结束就是end，在begin和end之间是sql语句块。用call +存储名来调用

我跟您讲一下中间这部分语句块吧，我先定义两个变量，i和y，y用来接收订单编号，默认为空，i是为了后续自增和循环使用的，默认值为1

用一个while循环，条件是n>i 这个n就是循环的次数。然后订单编号这里，我用到了左补位函数lpad，有三个参数，第一个是之前设置的i变量，第二个参数就是补位的位数，第三个参数就是补位的数。

就比如，i是2，那么他会自动在前面补5个0。然后通过concat拼接字符串，比如 深圳 。然后用insert into 表名 values（数据）插入数据。最后写个i=i+1实现订单编号自增长，同时配合while循环。

最后调用存储，然后去查表即可。

**10.你们数据库有多少张表**

我们数据库有两百多张表。

就比如说用户信息的表，表名是叫cus\_account, 里面我记得有cus\_account

Cus\_pwd，cus\_income ,cus\_expend，bank\_card等等

还有就是有张sys\_service表，有service\_name、service\_qq、service\_number、 service\_job、service\_mail 、joined\_city

备用：wf\_combo表，有combo\_code、combo\_name、update\_count、combo\_start\_time、combo\_end\_time

**拓展1：oracle和mysql的区别**

**拓展2：数据库测试思路**

**八、网络基础相关**

**1.http、https的区别**

http是明文传输；https是加密传输。

http连接简单，是无状态的；https是可以进行加密传输、身份认证的网络协议

http的端口是80；https的端口号是443

http可以免费申请；https需要到ca申请证书，需要交费

**2.get和post的区别**

get的请求参数写在URL中，post的请求参数写在body中。

url长度有限，get的请求参数写在url中可能会超出上限，但在post的body中没有这个限制

get请求常用于查询，post请求常用于提交操作

get相比于post安全性更低，因为请求参数就写在url中

get请求只能进行url编码，而post支持多种编码格式

get产生一个TCP数据包；post产生两个TCP数据包

**3.cookies和session的区别**

cookie

Cookies是一种技术，它能够把你在访问网站时产生的一些行为zhi信息给读dao取保存下来。

  （1）cookie是客户端的状态管理技术，cookie的时效性比较长。

  （2）cookie安全级别比较低。

  （3）cookieb有大小上限，但不会占用内存

（4）cookie保存在浏览器中

session

session是一次浏览器和服务器的交互的会话

  （1）session是服务端的状态管理技术，session的时效性比cookie短。

  （2）session的安全性比cookie更高

（3）session没有大小上限，但是会占用内存，影响性能

**4.TCP和UDP的区别**

1、TCP面向连接；UDP在发送数据前不需要建立连接

2、TCP相比UDP更加可靠。TCP传送的数据不丢失、不出错。而UDP只是尽最大努力交付，但不保证可靠交付

3、UDP时效较好，效率比TCP高，适用于高速传输和时效性较高的通信或广播通讯

4.每一条TCP连接只能是点到点的;UDP支持一对一，一对多，多对一和多对多的交互通信

5、TCP对系统资源要求较多，UDP对系统资源要求较少。

**5.http常见的状态码**

状态码一般分为五类，1开头表示请求已接受，2开头的表示请求成功，3开头的表示重定义，4开头的一般是客户端相关，5开头的一般是服务端相关

比如说，100就是请求已接受，客户应该继续发送请求其余部分

200就是一切正常

400就是请求错误、404指未找到资源，路径错误

500是服务器报错、502是网关无响应，503就是服务不可用，504网关超时

**6.http的请求头和响应头包括哪些信息**

请求头有：

请求方法、传输协议、域名、Content-Length（传输的正文大小或内容长度）、Connecttion（客户端与服务端连接类型）还有Content-Type（设置请求体的MIME类型），Accept-Encoding（接受的编码格式）、Accept-Language（接受的语言）、Cookie（设置服务器使用Set-Cookie发送的http cookie 其作用就是用于解决记录客户端的用户信息）、User-Agent：浏览器的用户代理字符串

Referer：发出请求的页面的URL

响应头有：

传输协议、服务器名称、Set-Cookie、Content-Type（设置响应体的MIME类型）、Transfer-Encoding（设置传输实体的编码格式）、Date（设置消息发送的日期和时间）

**7.七层模型、五层模型，每层对应的协议，协议的名称**

TCP/IP五层模型：

应用层，协议：HTTP Telent FTP TFTP DNS SMTP

传输层，协议：TCP UDP

网络层，协议：IP ICMP RIP IGMP

数据链路层，协议：ARP RARP IEEE802.3 PPP CSMA/CD

物理层，协议：FE自协商 Manchester MLT-3 4A PAM5

OSI参考模型（七层模型）

应用层、表示层、会话层 --------- 应用层

传输层 ----- 传输层

网络层 ----- 网络层

数据链路层 ----- 数据链路层

物理层 ----- 物理层

**8.说一说你对http的理解\认识**

http 的内容挺多的，我就站在测试的角度跟你讲一下，我在工作中之前接触到的一些东西

http是一种超文本传输协议，是明文传输的，http连接简单，是无状态的，它的端口号是80。

然后在http一些常见的状态码，分为五类，1xx：请求已接收，2xx：请求成功，3xx：重定向，4xx：客户端相关，5xx：服务器相关，比如100表示请求已接受，客户应该继续发送请求其余部分，200就是请求成功、400是请求错误、401是未授权访问、403是资源不可用、404是找不到指定位置的资源、500就是服务器报错、502是网关无响应，503就是服务不可用，504网关超时

讲http、https区别

GET、post区别

http请求头包括：。。。。。

**九、接口**

**1．一个接口包含哪些信息？**

url、接口协议、接口名称、传递方式、数据提交方式、接口功能描述、入参信息（参数名，入参说明）、响应消息（参数名，返回说明）

**2.你们公司接口什么时候测的？**

开发接口做好了，UI界面没出来之前测接口

**3.你觉得为什么要测接口？**

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）

**4.你接口测试过哪些bug？**

bug有很多，我给您举几个例子吧。

第一个例子。比如投资界面最低投资金额是1000，输入框前端做了限制大于1000。那么正常来说，我们测试的时候，也需要测试小于1000的情况下。很显然前端只能输入大于1000的，但是我当时用接口测试工具去测试的时候，就发现输入999时，页面显示成功，系统后台并没有识别到，而是正常的给出了返回值。那么，这就是一个后端接口bug了。

第二个例子。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。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。那么这也是一个bug，从这里可以检查服务端的功能是否OK。

第三个例子，错误提示处理不当，导致用户看到晦涩的错误码

**5.你在工作中负责过哪些接口的测试？**

我项目接口挺多的，比如用户管理接口、贷款接口、产品接口、订单接口、第三方接口等等，就拿贷款接口来说，又可以分为贷前、贷中、贷后三大模块。贷前我们要填写认证信息，调用信贷管理接口、风控调第三方征信接口。贷中，我们要调资金管理接口，资金下发要调银行接口。贷后，客户查看信息要调还款计划接口，还有流水接口、财务核对接口等等很多，还有储存客户信息后台还有第三方档案柜接口。

**6.接口自动化和UI自动化的区别？**

第一、接口自动化的覆盖率高于UI自动化

第二、接口自动化执行效率高于UI自动化的执行效率

第三、UI自动化是高投入，低回报；接口自动化地投入，高回报

**7.接口测试是站在用户角度去测还是后台角度去测？**

站在用户角度

**8.接口测试用例设计思路（你接口用例怎么设计的？）**

1、正常的流程。

2、根据参数的数据类型设计

3、根据参数的选填必填设计

4、根据参数的长度设计用例

5、异常的场景（场景法）

做接口测试用例之前，一般要考虑到以下几点：

  （1）接口的安全性：敏感信息是否有加密处理。

  （2）接口的性能：是否支持多用户的并发测试。

  （3）接口的基本功能是否正常，测试它是否符合接口文档。

  （4）接口对应的key：考虑key的个数、key的命名，key为空，修改key。

  （5）接口对应的value：值是否正确、值的长度、值的数据类型、空值等。

  （6）接口的依赖关系：测试相关联的接口、登入之后的接口。

  （7）cookie：cookie值的错误，失效的cookie，cookie为空看是否能进行登陆之后的操作。

  （8）接口涉及数据中的数据库操作，要验证数据库中的数据是否有变化。

**9.如何构建一个接口测试？**

python：

首先我会去看接口文档，拿到接口的url、接口数据提交格式、接口请求方式、入参等信息。然后在写用例之前，需要导入requests库，用requests库中的request方法构建，有三个参数，第一个填请求方式、第二个是接口url，第三个填写入参信息，然后就是右键run执行。接下来进行结果分析对比，根据接口返回的结果跟接口文档的response进行对比

jemeter：

首先我会去看接口文档，拿到接口的url、接口数据提交格式、接口请求方式、入参等信息。

接着在测试计划下面添加一个线程组，并在线程组下添加一个http请求，然后写入协议、服务器地址、端口、请求方法、资源路径、Content Encoding。接着添加查看结果树，添加响应断言和断言结果。然后点击运行，查看和分析结果，跟接口文档的response进行对比

**10.接口数据提交格式**

1.application/json

2.x-www-form-urlencoded

3.multipart/form-data

4.text/xml

**11.Python/jemeter如何构建https请求？（拓展）**

python：

jemeter：

用jemeter访问https接口，需要添加ssl层证书。首先导出证书，然后到jemeter中，添加线程组，再添加一个http请求，输入接口相关的数据，然后到jemeter选项里，点击SSL管理器，然后倒入证书即可。接下来点击运行，会提示输入密令，这个输入设置好的密令就可以了

**12.有没有测过测多接口和关联接口？业务上多个相关联的接口怎么处理？【登录之后怎么测？怎么拿cookie/session,要么做re表达式，讲怎么把返回的json转换，session方法里可以保存会话的方法】**

答：测过。在jmeter上选择配置文件添加http cookie管理器来保持当前窗口会话。关联接口，返回值为xml格式。通过正则表达式提取里面的值，作为下一个接口的入参。

在python里，登录后的接口通过request模块里调用cookie方法获取值作为下一个接口的入参。调用session方法保持同一会话。导入re（正则）模块中的findall方法，来获取里面的值，作为下一个接口的入参。

**13.当接口比较多的时候如何处理？  （封装）**

答:我先创建几个层配置层、构建层、用例层、运行层、报告层

用xlrd模块里面的方法可以获取Excel表格中保存的测接口的需要用到的参数，这里需要注意的Excel表格中的数据我们会在表格第一行做一个数据分类然后写入对应的数据，我们在Python中获取数据时，就可以单独获取第一行的内容做为键和后面获取的值来一一对应，将获取到的值赋值给变量data。

 Ddt数据驱动，将我的数据全部放置到一个表格里面，然后通过xlrd的方法读取本地表格的数据形成对应的一个键值对数据，配置一个变量接收，传入到接口请求中去，形成报告HTML

**Python+ddt**

**python + ddt**

第一步、接口数据构建在我们的表格中

a、表格的第一行是对应的接口入字段信息（请求方式、body字段、url等）

b、表格从第二行开始对应上的是一个接口的入参信息，每一行是一个完成的接口入参

第二步、读取我们的本地表格中的数据

a、需要的一个模块去读取表格中的数据：xlrd，往表格中写入数据：xlwt--可不讲

b、读取表格中第一行作为key，--以列表的形式返还通过循环去读取表格中的其他行的内容作为values----以列表的形式返还，然后通过zip方法讲key和values组装成字典，每个一个字典都是一个完成的接口入参

第三步：构建接口请求

该请求是用来等待后续ddt数据驱动将数据传入，根据对应的数据决定是get还是post请求

第四步：ddt数据驱动层

常见的几个装饰器

1、类装饰器：ddt

2、方法装饰器：data：包含多个你想要传给测试用例的参数

3、unpack：通常data中包含的每一个值都会作为一个单独的参数传给测试方法，

如果这些值是 用元组或者列表传进来的，可以用unpack方法将其自动分解成多个参数url等）

**14.接口自动化讲解**

我们公司之前做接口自动化是通过python+requests来实现的，为了方便后期对用例的管理，我们采用了模块分层设计，提高了脚本的一个可维护性和利用率。我们会分为以下几层：配置层、组建接口层、用例层、执行层、报告层。配置层是用来存放接口参数信息的，组建接口层里就是写组建接口的脚本。

  那我就拿我上一个项目里的借款接口为例，我们借款的功能是向服务器提交数据进行请求，我们采用的请求方式时post，借款接口的参数信息有对应的url、以键值对形式存在的data信息，像是借款人姓名、借款金额这些入参信息，我们可以从接口文档中获取。

  之后就是组建接口，这个是在组建接口层编写，导入requests库和配置层。接着定义一个类，把接口请求的流程封装，便于后续其他模块调用。然后在构造函数中，通过requests.Session()定义一个session对象，用来保持会话状态。再通过session对象去组建请求。然后使用requests的post方法，把url和入参信息从配置层取出并写入，再用一个变量来存响应体内容。比如r变量。然后使用r.json()把响应体内容转换为字典的格式，便于后续断言。

  接下来就是在用例层编写自动化用例。在这块我们是用了unittest框架，对用例进行组织管理。首先导入unittest库和组建接口层，接着定义一个类，继承unittest下的TestCase。用setUpclass和tearDownclass这两个方法实现对测试环境的初始化和还原。Setupclass中就是调用组建接口层的类，给他生成一个对象r。然后就是定义实例方法，编写用例。然后用一个变量去接收返回的响应体内容，根据内容，用assert去进行断言，看接口是否调通。

接下来就是执行模块，导入一个HTMLTest

runner，定义好报告保存路径，以及用例层路径，通过Testloader下的discover方法搜索并整合用例，用HTMLTESTRUNNER方法，写入报告相关信息，如title，用例执行情况，测试者。用run方法去执行用例。

最后到报告的路径下，查看并分析测试报告，这就是大概的一个流程

**15.你们平台有没有做过类似支付的平台？**

充值怎么充：储存卡、信用卡直接充到余额中。一旦说了做过，要准备讲一下支付的流程

**16．你能不能具体讲一下支付流程？**

 微信支付宝举例：我们这边前台下单，就会调用后台接口生成一个订单给到前台，前台根据这个信息调用支付宝支付接口，支付后支付宝后台判断支付宝结果回调给支付宝前台，前台将信息给到用户，支付宝后台直接将信息给到商家后台

我们项目中有充值的接口，用户要充值的话

   余额：

我就以当时做的金融借贷平台为例来讲一下余额支付，因为这个在借贷平台中用的比较广泛，也比较方便。首先用户在平台里面肯定会开通一个第三方存管账户，这个账户的功能主要是对平台来往资金安全起到一个监管的作用，同时还会开通个人账户，它主要就是对我们个人的一个余额进行管理。当我们账户余额不够的时候，我们就可以充值，那么这个时候就会用到第三方存管账户了。当我们要出借的时候，点击出借按钮，填写出借的信息，会形成出借的订单。点击确认后，订单将发送平台的后台，后台进行初步审核，并返回一个支付页面的信息到用户手中。用户看到信息后，选择余额支付，并输入支付密码，形成支付订单并发送给后台。后台接收后，会对用户账户余额和出借金额进行判断，当用户账户余额大于出借金额，则判断通过。此时，后台会调用财务系统接口，由财务系统对该支付订单进行一个记账，并返回一个记账单给到后台。后台再形成一个支付结果，并返回给用户。到这里，展示在用户眼前的就是显示支付成功的页面。

二维码支付：

用户选择业务订单后，页面会跳转到支付网关，支付网关返回一个选择支付渠道的页面。用户选择支付渠道后，又将结果给到支付网关。支付网关调用支付系统进行处理后，再向资金渠道发起二维码支付请求。资金渠道向第三方支付机构请求二维码支付信息。第三方支付机构处理后返回二维码信息给资金渠道。资金渠道再返回给支付系统，再返回给到用户、。此时在用户眼前展示的就是二维码。用户进行扫描支付，该请求直接发送到第三方支付机构。第三方支付机构接收处理后，将结果返回给资金渠道，资金渠道调用财务系统进行记账。财务系统将记账结果返回到支付系统，再由支付系统给到用户。最后，呈现在用户眼前的就是支付结果的页面。这样，就完成了二维码支付的流程。

**支付的测试点**

记两三个支付流程

1、订单是否可以正常提交

2、同一订单是否可以重复提交

3、提交的订单能否取消

4、后台收到订单能否进行保存

5、支付接口收到订单信息是否和提交订单信息一致

6、点击确认支付，页面能否跳转至选择支付接口页面

7、返回的订单页面，是否有订单信息

8、点击确认支付，能否选择微信支付

9、点击确认支付，能否选择支付宝支付

10、点击确认支付，能否选择网银支付

11、输入错误的密码能否支付成功

12、不输入密码能否支付成功

13、输入账号和密码不相符，能否支付成功

14、连续多次输入错误密码，是否还能继续支付

15、无网络状态下，能否扣款成功

16、弱网状态下，能否扣款成功

17、点击多次提交支付，能否多次扣款

18、余额不足情况下

19、扣款金额是否和订单金额一致

20、后台商家能否收到支付成功消息

21、支付成功后，页面是否跳转至指定页面

22、支付成功后，订单状态是否更新为已付款

**17.第三方接口怎么测的**

有测试过，我的极光金融项目中的借款在放款之后会进行电子档案的归档入库，将借款相关的档案存储在第三方的智能档案柜中，这个项目有两个第三方接口需要测试，一个是归档的入库接口，另一个是查询档案状态的接口。测试过程是这样的，我们接口文档会提供需要传入第三方接口的相关参数信息（参数：必填：姓名username，身份证号id，借款金额loan Amount，借款期限loanPeriod、还款方式repayment、手机号userMobile，直系亲属的姓名relativeName、手机号relativeMobile、关系kinship===选填：个人工作单位workUnit、薪资情况salary、车牌号码carNum），第三方接口是不需要我们测的，我们只需要调用这个接口，这个接口是第三方已经测好存在的，然后我们项目组交钱，第三方将接口提供给我们使用。我们会进行验收测试，根据参数的选填必填、数据类型、长度限制进行接口用例设计，运用postman这个工具选择post方式，在body模块按照用例传入归档的测试参数，点击send进行测试。

**十、抓包**

**1.你有没有用过抓包工具？**

用过，我用过fiddler工具抓包,还有浏览器中自带的F12工具

**2.讲一下fiddler工具的原理**

fiddler是位于客户端和服务器端之间的HTTP代理， 它能够记录客户端和服务器之间的所有 HTTP(S)请求，可以针对特定的HTTP(S)请求，分析网络传输的数据，还可以设置断点、修改请求的数据和服务器返回的数据。

fiddler在浏览器与服务器之间建立一个代理服务器，Fiddler工作于七层中的应用层，能够捕获通过的HTTP(S)请求。Fiddler启动后会自动将代理服务器设置成本机，默认端口为8888。Fiddler不仅能记录PC上浏览器的网络请求数据，还可以记录同一网络中的其他设备的HTTP(S)请求数据。

**3.fiddler常见的功能有哪些？**

1.可以抓http和https的包

2.可以用来做接口测试

3.可以打断点

4.指定一个url进行抓包

5.可以抓取手机的包

**4.fiddler怎么打断点？**

分为两种，第一种是全局断点，第二种是单个断点

全局断点就是中断fiddler捕获的所有请求，在rules-> automatic breakpoint  ->before requests响应前（after responses，响应后）

单个断点就是给某个网址单独打断点，输入命令就可以了， bpu + url

响应后的断点就是bpafter + url

取消断点，输入bpu即可

**5.fiddler如何抓https包？**

点击tools菜单，选择Fiddler Option，然后选择HTTPS选项，勾选decrypt HTTPS traffic选项，然后弹出一个框点击yes即可，后续就可以抓https的包了

**十一、PYTHON**

**1.列表、字典、元组的区别**

列表和元组是有序集合，字典不是

列表是可变对象；元组是一个不可变对象；字典中的键是不可变的，但值可变

列表和元组支持索引和切片，而字典不支持

列表的表达符是[]，元组的表达符是(),字典的表达符是{}

**2.字典列表元组中常见方法？**

这个自己复习即可

**3.python中的模块？你用过什么？里面有哪些方法**

有time模块、random模块、os模块、re正则模块

time里面有time（1970至今的秒数）、ctime（固定格式当前时间）、sleep（休眠，单位是秒）、strftime（时间格式化输出）

random模块里有random（随意生成0-1之间的浮点数，不包括1）、randint(x,y) 随机生成指定范围内整数，包括xy、sample(序列,n)从序列中随机取n个独立元素、choice随机生成字符

os模块有 path.isdir()判断是否为目录、path.exists()判断文件或目录是否存在、listdir(dirname)列出指定目录下的目录或文件、path.split分割文件名与目录、mkdir、path.join、rename(old,new)

re模块有match(第一个字符开始匹配，如果第一个字符不是要匹配的额类型，匹配失败报错)

serch(从第一个开始找、一找到就返回第一个字符，找不到就往下找，找不到就报错)

findall(从第一个开始找，找到全部相关匹配为止，找不到返回一个空列表)

complie(编译模式生成对象，找到全部相关匹配为止，找不到返回空列表)

**4.内置函数？**

def 定义一个函数 \*name可变长元组 \*\*name可变长字典

return、continue、break、format

zip() 可将对象中的元素打包成一个个元组，返回由元组组成的列表。

open()

**5.python熟悉吗？（别说熟悉）**

也不能说熟悉，我用python做过自动化 ，基本的一些还是了解的。

**十二、UI自动化**

**1.你做自动化多久了？（你什么时候开始做自动化的？）**

最早接触自动化是在我做第一个项目那会，那时接触到了一些，了解到了一些自动化知识。

**2.你们自动化是用什么做的？**

我们公司是用python+unittest框架做的。

**3.你们公司哪些地方做过自动化？**

一般是功能比较稳定，需要频繁回归测试的业务，还有页面元素不会变动的地方需要做自动化。就比如我上个项目中的个人中心，页面元素已经固定了，这块就做了自动化，还有信息披露、我要投资的页面也做了UI自动化

**4.UI自动化的原理是什么？**

我们自动化是python+selenium做的，主要就是基于selenium中的webdriver。

比如说用的谷歌浏览器

通过webdriver去调用谷歌浏览器的类，通过类去调用谷歌浏览器的驱动（驱动放在Scripts下），谷歌驱动就会驱动浏览器，通过元素定位的方法定位元素，从而实现自动化

**5.公司自动化是你负责的还是跟你同事一起做的？**

是我一个人负责的。我们项目有web端和app端，我主要负责web端的测试，所以ui自动化也是我负责的

**6.自动化脚本设计思路**

自动化测试适用范围主要是回归测试针对回归测试来设计自动化测试脚本的思内路：

1.确定容是否需要进行自动化测试

2.需要自动化测试，则确定哪些内容需要进行自动化测试

3.选取需要自动化测试的内容相关的测试用例

4.根据这些用例，构建数据文件

5.根据数据文件设计相应的逻辑，以方便数据文件的输入为主

6.自动化测试脚本应该包含以下内容：

1.可重复的操作步骤

2.可替换的输入

3.具有比对预期输出与实际输出的功能4能够自动将比对的结果输出到一个统一的文件上

**7.关于定位场景**

**隐藏列表定位：**

from selenium.webdriver.common.action\_chains import ActionChains

test=driver.find\_element\_by\_id('s-usersetting-top')

ActionChains(driver).move\_to\_element(test).perform()

time.sleep(1)

driver.find\_element\_by\_link\_text('搜索设置').click()

time.sleep(1)

driver.find\_element\_by\_xpath('//\*[@id="se-setting-7"]/a[2]').click()

tt=driver.switch\_to.alert

time.sleep(1)

tt.accept()

**弹框定位：**

（1）iframe框，先通过元素定位到iframe框，然后使用switch\_to.iframe(iframe框定位或索引值，0开始)

退出iframe框：driver.switch\_to.default\_content()

（2）警告框,t=driver.switch\_to.alert()

accept()点击确定

t.dismiss()点击取消

send\_keys() 输入型弹框可以在弹框上进行文本输入

**下拉滚动条定位：** t='window.scrollTo(0,5000)'

driver.execute\_script(t)

**下拉框：**需要导入模块 selenium.webdriver.support.select import Select

定位可以通过元素去定位，例s1=driver.find\_element\_by\_id('xxxx')。

下拉框内容可以用 Select(s1).select\_by\_index(3)，索引定位

或Select(s1).select\_by\_values(3)，通过value定位

或Select(s1).select\_by\_visivle\_text(‘文本内容’) ，通过文本内容定位

**关闭窗口：**close()

**关闭浏览器：**quit()

**8.元素定位方法有哪些（9种）**

元素定位方法的话，主要有以下几种

id,name,class,link\_text,partial\_link\_text,js定位,tag\_name,xpath,css

**9.等待的方式有哪些（3种）**

**强制等待**

sleep

**隐式等待**

driver.implicitly\_wait(5)

针对所有元素，设置等待时间，在规定时间内出现则继续向下执行，否则报错

**显式等待**

WebDriverWait(driver,5,0.5).until(EC.presence\_of\_element\_located(By.ID,'kw'))

.sendkeys(u'多测师')

是单独针对某个元素，设置一个等待时间如5秒，每隔0.5秒检查一次是

否出现，如果在5秒之前任何时候出现，则继续向下，超过5秒尚未出现则抛异常

**10.断言的方式**

assert\*():一些断言方法：在执行测试用例的过程中，最终用例是否执行通过，是通过判断测试得到的实际结果和预期结果是否相等决定的。

assertEqual(a,b，[msg='测试失败时打印的信息']):断言a和b是否相等，相等则测试用例通过。

assertNotEqual(a,b，[msg='测试失败时打印的信息']):断言a和b是否相等，不相等则测试用例通过。

assertTrue(x，[msg='测试失败时打印的信息'])：断言x是否True，是True则测试用例通过。

assertFalse(x，[msg='测试失败时打印的信息'])：断言x是否False，是False则测试用例通过。

assertIs(a,b，[msg='测试失败时打印的信息']):断言a是否是b，是则测试用例通过。

assertNotIs(a,b，[msg='测试失败时打印的信息']):断言a是否是b，不是则测试用例通过。

assertIsNone(x，[msg='测试失败时打印的信息'])：断言x是否None，是None则测试用例通过。

assertIsNotNone(x，[msg='测试失败时打印的信息'])：断言x是否None，不是None则测试用例通过。

assertIn(a,b，[msg='测试失败时打印的信息'])：断言a是否在b中，在b中则测试用例通过。

assertNotIn(a,b，[msg='测试失败时打印的信息'])：断言a是否在b中，不在b中则测试用例通过。

assertIsInstance(a,b，[msg='测试失败时打印的信息'])：断言a是是b的一个实例，是则测试用例通过。

assertNotIsInstance(a,b，[msg='测试失败时打印的信息'])：断言a是是b的一个实例，不是则测试用例通过。

**11.UI自动化怎么截屏？（拓展）**

使用

dr.get\_screenshot\_as\_file（路径+图片名.png）

**12.setup和setclass的区别**

setup绑定的是类的方法，setupclass绑定的是类。Setup是每一个实例方法之前都执行一次，setupclass是所有实例方法执行前执行一次，只执行一次。

**13.你做UI自动化遇到过什么问题，你怎么解决的？**

做自动化遇到的问题也有，虽然做了一段时间了，用例也写了不少。我记得我刚学自动化那会，经常会定位不到一些场景或元素，现在比较少了，最近自动化遇到的问题也有。就是自动化本写的多了，执行效率就变低了很大，我是对用例分层分模块去执行，还有就是优化了一下脚本，用显示等待去优化那些无效等待的场景（我只需要其中一个元素，正常来说需要等待所有元素加载完，但是我不需要其他元素，所以可以用显式等待，当我需要的那个元素出现就去执行）

还有就是遇到过之前还可以执行的脚本，后面报错，这个一般就是元素定为问题，有的元素的标签不唯一，一般默认是找到第一个。向上查看，元素是否在frame或iframe框架中，检查元素属性是否会变动，是否是隐藏的

**14.自动化脚本维护是你做的吗，怎么维护？**

是的。为了方便维护我会对脚本进行封装。还有就是命名的方式，要见名知意，还有就是要记得写注释。

要尽量减少循环和循环嵌套，写代码简介高效的话，运行效率也高，还有就是合理使用等待方式

**15.你做自动化的成果**

对用例的优化、减少了循环

**16.脚本的设计是你设计的吗，怎么设计的（框架是你写的吗）**

是的。

为了方便后期管理，我一般会把用例进行分层处理，分为以下几层：配置层（config）、数据层、公共层、用例层、报告层、执行层。配置层中放的就是其他层的路径信息，数据层中存放的是测试需要用到的数据（[test\_data]标题），公共层中放的就是一些常用的方法，比如读取数据层的方法，还有一些元素定位方法、窗口切换方法、页面元素集合等。用例层放的就是我们编写的测试用例，报告层用来存储测试报告、执行层就执行用例并生成报告。

我拿我们项目里我要投资来给你举个例子，在用例层中编写，首先定义一个类继承unittest下的Testcase，一些前置条件写在setupclass中tearDownclass用于执行完成后进行资源释放。接下来从公共层调用登录方法，然后需要点击我要投资模块，

**17.具体介绍一下unittest框架**

unittest是python自带的一个库，后期我们写的用例越来越多，我们要考虑用例编写的规范和组织，便于后期维护，而unittest下有很多模块可以帮助我们封装用例。

就比如TestCase类，一个TestCase的实例就是一条测试用例。TestSuite类可以帮助我们把很多测试用例集合在一起执行。TestLoader可以自动搜索以test开头的用例。TextTestRunner可以用来执行测试用例，其中的run(test)用来执行TestSuite/TestCase。TestFixture里有测试准备前要做的工作和测试执行完后要做的工作.包括setUp()和tearDown()。通过覆盖TestCase的setUp和tearDown来实现。

在unittest提供了丰富的断言方法，例如assert、assertequal、assertnotequal、assertTure、assertFlase、assertin、assertnotin等

在使用unittest框架的时候，关于用例的命名，要以test开头，并且用例的执行顺序是根据ASCII码表的顺序来执行的。

**拓展**

一、unittst框架的作用

当我们写的用例越来越多时，我们就需要考虑用例编写的规范与组织，以便于后期的维护，而unittest正是这样一款工具

二、unittest是Python自带的标准库中的模块，其中包括：

1、TestCase类

2、TestSuite类

3、TestLoader类

4、TextTestRunner类

5、TextTestResult类

6、TestFixture类

解释下：

TestCase：

一个Testcase的实例就是一个测试用例，测试用例就是一个完整的测试流程，包括初始化setUp、运行run、测试后的还原tearDown

TestSuite：

对一个功能的测试往往需要多测试用例的，可以把多的测试用例集合在一起执行，这就是TestSuite的概念。常用addTest()方法将一个测试用例添加到测试套件中

TextTestRunner：

是用来执行测试用例的，其中的run(test)用来执行TestSuite/TestCase。测试的结果会保存在TextTestResult实例中

TestFixture：

测试准备前要做的工作和测试执行完后要做的工作.包括setUp()和tearDown()。通过覆盖TestCase的setUp和tearDown来实现。

TestLoader：

是用来搜索所有以test开头的测试用例，然后将其加入到testsuite中

**18.你对python+selenium掌握的程度？（你当前的自动化水平如何？）**

什么水平我也不好说，自动化环境我搭建过、脚本也有设计、编写、执行过，分层封装我都做过，如果让我来独立负责自动化我也有信心做好，当然可能也会遇到一些问题，我会自己想办法，独立去解决。

**19.Unittest和pytest的区别？（拓展冷门）**

一、用例设计规则

1.unittest

（1）测试类必须继承unittest.TestCase

（2）测试函数必须以"test\_"开头

（3）测试类必须有unittest.main()方法

2.pytest

（1）测试文件的文件名必须以"test\_"开头，或者以"\_test"结尾

（2）测试类命名必须以"Test"开头

（3）测试函数名必须以"test"开头

测试类里面不能使用"\_\_init\_\_"方法

断言的区别：

Unittest断言丰富多而pytest就一个assert

用例前置和后置

1.unittest前置和后置

（1）通过setup每个用例执行前执行，teardown每个用例执行后执行

（2）通过setupclass类里面所有用例执行前执行，teardownclass类里面所有用例执行后执行

2.pytest前置和后置

（1）模块级别：setup\_module/teardown\_module，整个.py全部用例开始前执行/全部用例执行完后执行

（2）函数级别:setup\_function/teardown\_function，只对函数级别生效，每个用例开始前和结束后执行一次

（3）类级别：setup\_class/teardown\_function,只对类级别生效，类里面所有用例开始前执行一次，所有用例执行完执行一次

（4）方法级别：setup\_method/teardown\_method,只是类里面方法级别生效，方法开始前执行一致，方法结束后执行一次

生成报告方式

1.unittest

通过HTMLTestRunner生成

2.pytest

（1）通过pytest-html生成html格式报告

（2）通过allure生成方案（很详细）

**测试数据放哪？**

1.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，写的配置文件里（如ini） 2.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 3.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,excel都可以 4.对于可以反复使用的数据，比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 5.对于邮箱配置的一些参数，可以用ini配置文件 6.对于全部是独立的接口项目，可以用数据驱动方式，用excel/csv管理测试的接口数据 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py脚本的开头，十年八年都不会变更的

总之不同的测试数据，可以用不同的文件管理

**十三、主观题**

**1.离职原因（必问）**

第一、从发展角度讲（公司角度，疫情原因，没有太多业务，对我来说）

第二、从薪资角度（去年年底说提薪资，今年疫情影响，一直没有落实下来）

第三、个人原因（）

**2.职业规划（必问）**

现在我也没有太长远的规划，就近期来看，如果我能加入贵公司，我的规划就是尽快融入贵公司，胜任测试工作。我觉得测试这个行业要求在变高，竞争也越来越大，所以还是要提升自己，先是要巩固完善自己目前会的技能，然后对自己技术薄弱环节进行提升，像是我在性能、安全方面的经验稍微有些不足，这方面我会去学习提升。如果从理想来说，如果可以的话，未来我想往管理或产品方面发展，这就是我的个人规划。

**3.你对加班怎么看（必问）**

你经历不对吧，加班加的（滑稽）

我们之前加班134规定加到9点，周末看情况，有时候加一天，有时候加两天，这个要看忙不忙

晚上八点有餐补，十点有车费补贴

**4.如何看待加班**

两种情况：1.工作效率不高，导致无效加班

2.特殊情况，紧急发版本，这个要积极配合加班

一般遇到紧急发版本，这个要积极配合加班，还有就是在交付和上线的时候加班是必然的

还有就是因为工作效率不高导致的无效加班

**十四、性能**

**1.你有没有做过性能？你们公司用什么做的？**

做过，用jemeter针对接口的性能测试。

**2.性能测试流程？**

性能需求分析（哪些场景需要做性能、性能的标准）

性能测试环境准备

性能脚本开发和准备

性能测试执行

性能测试结果分析

性能调优

总结报告

**3.你们公司什么时候开始做的性能？**

根据性能需求，针对需要进行性能测试的业务场景，功能测试通过后就可以性能介入。

**4.你们公司什么地方做过性能测试？（项目哪些场景做过性能测试）**

第一、用户量比较大，业务使用频繁的场景需要做性能测试

第二、核心、重要的模块需要做性能测试

例如：我们项目的个人中心、首页、登录、注册、查询相关的接口

一般是在用户量比较大，业务使用比较频繁的场景，还有项目中的核心、重要的模块要做性能

就比如：我们项目中的个人中心、首页、登录、注册、查询相关的接口

**5.性能测试分哪几类？**

基准测试，负载测试、压力测试、容量测试、并发测试、可靠性测试、稳定性测试

**6.性能测试需要关注的指标？**

需要关注的指标可以分两类，一个是系统指标，一个是资源指标。

系统指标就是：TPS、最大响应时间、最小响应时间、平均响应时间，事务通过率

资源指标就是：CPU、内存、网络读写、存储

TPS（服务器每秒处理事务数）

一个系统在处理一个业务的时候，花了多少时间，是系统指标。

**7.你们公司做测试的时候，性能指标值是从哪来的？**

性能需求文档。或是业务规定的。或是根据线上的用户数进行评估得出的

这个的话我们会有一个性能需求文档，需求里有指标值，不过我想这个值应该是根据线上的用户数进行评估得来的

**8.有没有用过jemeter的插件**

记下jemeter用过的插件

用过，就比如说有

Bytes Throughput Over Time:可以吧不同时间吞吐量显示在（图表）聚合报告里

Hits per Second:每秒点击量

PerfMon Metrics Collector：服务器性能监控件，包括CPU、Memory，Network，I/O等等

Transactions per Second：每秒事务数，服务器每秒处理的事务数

**词汇解释**

1. 吞吐量

指在一次性能测试过程中网络上传输的数据量的总和。

对于交互式应用来说，吞吐量指标反映的是服务器承受的压力，在容量规划的测试中，吞吐量是一个重点关注的指标，因为它能够说明系统级别的负载能力，另外，在性能调优过程中，吞吐量指标也有重要的价值。如一个大型工厂，他们的生产效率与生产速度很快，一天生产10W吨的货物，结果工厂的运输能力不行，就两辆小型三轮车一天拉2吨的货物，比喻有些夸张，但我想说明的是这个运输能力是整个系统的瓶颈。

提示，用吞吐量来衡量一个系统的输出能力是极其不准确的，用个最简单的例子说明，一个水龙头开一天一夜，流出10吨水；10个水龙头开1秒钟，流出0.1吨水。当然是一个水龙头的吞吐量大。你能说1个水龙头的出水能力是10个水龙头的强？所以，我们要加单位时间，看谁1秒钟的出水量大。这就是吞吐率。

负载

对被测系统不断施加压力，直到性能指标超过预期或某项资源使用达到饱和，以验证系统的处理极限，为系统性能调优提供依据；

并发

①狭义上的并发：所有用户在同一时间点进行同样的操作，一般指同一类型的业务场景，比如1000个用户同时登陆系统；

②广义上的并发：多个用户与系统发生了交互，这些业务场景可以是相同的也可以是不同的，交叉请求和处理较多；

压力

系统在一定饱和状态下，例如CPU、内存等饱和情况下，系统能够处理的会话能力，以及系统是否会出现错误

特点：主要目的是检查系统处于压力情况下应用的性能表现，重点在于有无出错信息产生，系统对应用的响应时间等

事务

性能测试中，事务指的是从端到端，一个完整的操作过程，比如一次登录、一次筛选条件查询，一次支付等；

吞吐量

指在一次性能测试过程中网络上传输的数据量的总和，也可以这样说在单次业务中，客户端与服务器端进行的数据交互总量；

对交互式应用来说，吞吐量指标反映服务器承受的压力，容量规划的测试中，吞吐量是重点关注的指标，它能够说明系统级别的负载能力，另外，在性能调优过程中，吞吐量指标也有重要的价值；

吞吐率

吞吐量/传输时间，即单位时间内网络上传输的数据量，也可以指单位时间内处理客户请求数量，它是衡量网络性能的重要指标。

通常情况下，吞吐率用“字节数/秒”来衡量，当然，也可以用“请求数/秒”和“页面数/秒”来衡量；

TPS

Transaction Per Second：每秒事务数，指服务器在单位时间内（秒）可以处理的事务数量，一般以request/second为单位；

PS：看到很多博客或性能测试人员将QPS和TPS混为一谈，个人认为，他们是以测试结果的统计得到该结论的；

QPS是查询，而TPS是事务，事务是查询的入口，也包含其他类型的业务场景，因此QPS应该是TPS的子集！

QPS

Query Per Second：每秒查询率，指服务器在单位时间内（秒）处理的查询请求速率；

PS：TPS和QPS都是衡量系统处理能力的重要指标，一般和并发结合起来判断系统的处理能力；

**9.性能测试是怎么测的？**

在测试之前首先要了解它的准出原则：TPS、响应时间、并发数、吞吐量等分别是多少

我们公司是用jemeter做的性能测试，那我就拿我之前做的一个场景来说吧，那是从登陆到进入个人中心查看用户信息这么一个流程。目标是对服务器进行压力和稳定性测试，以及对CPU、内存的一个压测。

首先的话，在jemeter中创建一个线程组，添加http请求，填写相关信息，url、端口号等等，根据文档上写的我们是做五百用户的一个并发 ，把线程数设置为500，加载时间设为10s，模拟500个用户同时操作。还需要添加定时器，有两个参数,第一个是并发数，设置为500，第二个是时间，设置时间为10000ms。为了查看测试结果，可以添加聚合报告，聚合报告一般观察平均值、90%、95%用户请求时间、最大最小请求时间、错误率，还可以借用其他插件生成曲线图，基本上就是这样的

如果不符合，那就需要根据聚合报告、去找一下性能瓶颈，在根据瓶颈进行分析，给出调优方案，在最后完成一下性能回归测试 。

**10.你还记得你之前做并发的时候聚合报告的具体数据吗？你觉得合理的和不合理的有哪些？**

我记得当时用1000个用户同去进行对首页进行压测，平均响应时间是2S多，最小是150多毫秒

最大的有3s多

**11.你们性能测试是在测试环境还是真实环境测的（你们性能数据从哪里来）**

数据可以直接从数据库里拉。

我们公司做性能测试有一套专门的环境，是一个几乎和真实环境一样的滑稽，有专门的人搭建，维护的话也不是我们测试去维护的

我们老大让我去哪个环境测，我就去哪个环境测

**12.性能测试的结果你怎么分析？**

第一、看聚合报告，看响应时间（最大最小、平均时间）是否在合理范围内，以及出错率是否在性能规定的指标范围内

第二、从真正的性能指标着手，CPU、内存、网络读写等。看事务（登录、注册等）的通过率是否达到相应的要求，有意义的场景就可以当成一个事务。一千个人对首页并发，那就是一千个事务。

**13.性能调优会从哪几个方面去做？**

性能调优的话可以从以下几个角度去考虑

第一、可以从业务角度去考虑，比如说一个简单的场景，本来一个页面可以解决，但是把业务复杂化了，要用三个页面解决

第二、就是代码的架构，去考虑性能调优

第三、就是硬件方面去考虑，不过这个的话一般百分之八九十的性能问题都是人为造成，只有10%是因为硬件导致

第四、就是网络角度，可能网络达不到要求造成了延迟

第五就,是可以从连接池去调优，可用的连接数太少，造成请求等待

第六、就是数据库，添加索引查询数据、还有读写分离 ，就是主数据库负责增、删、改，从数据库来进行查的操作

**14.你在公司测试的时候有没有遇到过的性能问题（记）**

正常来说TPS指标要求是20/300ms，但是

问题一：压力测试中TPS一直上不去

网络带宽

分析思路：

在压力测试中，有时候要模拟大量的用户请求，如果单位时间内传递的数据包过大，超过了带宽的传输能力，那么就会造成网络资源竞争，间接导致服务端接收到的请求数达不到服务端的处理能力上限。

连接池

可用的连接数太少，造成请求等待。连接池一般分为服务器连接池（比如Tomcat）和数据库连接池（或者理解为最大允许连接数也行）。（关于连接池的具体内容，可参考之前的博客：性能测试：连接池和线程）

4、数据库配置

高并发情况下，如果请求数据需要写入数据库，且需要写入多个表的时候，如果数据库的最大连接数不够，或者写入数据的SQL没有索引没有绑定变量，抑或没有主从分离、读写分离等，就会导致数据库事务处理过慢，影响到TPS。

6、硬件资源

包括CPU（配置、使用率等）、内存（占用率等）、磁盘（I/O、页交换等）。

7、业务逻辑

业务解耦度较低，较为复杂，整个事务处理线被拉长导致的问题。

一、容量（压力）测试过程中cpu过高----这个要考虑压了多少用户

结果：cpu却到了80%多，指标是不能超过60%。

解决思路：如果是cpu过高，先把消耗cpu最多的进程找出来（top命令），再找到该线程下消耗cpu过高的是哪几个线程，再把该线程处理掉

二、内存溢出（堆溢出、栈溢出、持久代溢出）

解决思路：1、调整堆内存参数，一般是增加堆内存

2、减少批处理数据量

三、线程死锁：容量（压力）测试压测一段时间后，报连接超时

解决思路：

1、造成这种现象的原因很多，比如带宽不够，中间件线程池不够用，数据库连接池不够，连接数占满等都会造成连接不上而报超时错误

2、找到死锁的线程，分析对应的代码

四、数据库死锁

解决思路：

1、造成这种现象的原因很多，比如带宽不够，中间件线程池不够用，数据库连接池不够，连接数占满等都会造成连接不上而报超时错误。

2、（重点）数据库日志中搜索block，能搜到block的话就是存在数据库死锁，找到日志，查看对应的sql，优化造成死锁的sql。

五、数据库连接池不释放

上一次是通过日志分析或者查看代码，数据库连接部分是不是有创建连接但是没有关闭连接的情况。基本就是这种情况导致的，修改代码即可。

六、事务通过率不达标

保存单据的事务通过率总是80%，离99%差很远。

1.分析应用的日志，并没有发现报错信息。

2.分析中间件gc日志和数据库报告，压力并没有过载。

3.分析loadrunner报告，发现302的请求比较多。

陷入僵局。

1.在代码的保存方法中打印出语句。发现1000个请求，只有800个打印语句。

2.分析access.log，发现有1000个请求，其中有200个请求返回状态码是302，其他的是200。说明请求没有执行到保存的方法就被重定向了。

现在问题比较明确，找到被重定向的原因：

在loadrunner的机器上安装fiddler，捕获整个请求过程，最终定位到302的原因是请求。

HTTP/1.1 302 Moved Temporarily

X-Powered-By: Servlet/3.0 JSP/2.2 (InforSuite APP Server Java/Sun Microsystems Inc./1.6)

Server: InforSuite APP Server

Location: <http://10.10.81.91/web/gg/PustOutLogin.jsp>

Content-Type: text/html;charset=GB2312

Content-Language: zh-CN

Content-Length: 194

Date: Fri, 26 Oct 2018 01:47:51 GMT

是会话失效导致用户被登出。

jemeter参数化怎么做的？

两种方式本地CSV，或者用户参数

**十五、APP**

**1.你们app自动化是怎么做的？**

我们公司APP自动化是用python+appium做的，为了方便后续维护，我们是用unittest框架做的

**2.app自动化的定位方式？**

id、name、class、link\_text、partial\_link\_text、javascript（js）、xpath、css、tag\_name

**3.APP手机上下左右滑动的方法怎么封装的**

python-appium手机自动测试如何实现滑动呢，在webdriver 的api中只提供了指定两个坐标的滑动，但不同的手机分辨率不同，这样写出来的自动化脚本兼容性就不是那么好了。。那么怎么实现滑动呢，仔细看了下webdriber中的方法，发现有个方法get\_window\_size() ，这个方法可以获得手机的宽度和高度，这样我们就可以实现按屏幕比率滑动了。。具体方法如下：

首先我们给定固定三个参数：0.5  0.25  0.75，(参数合理就好)因为不管向上向下还是向左向右滑，他们的参数都在由大到小变化或者由小到大的变化

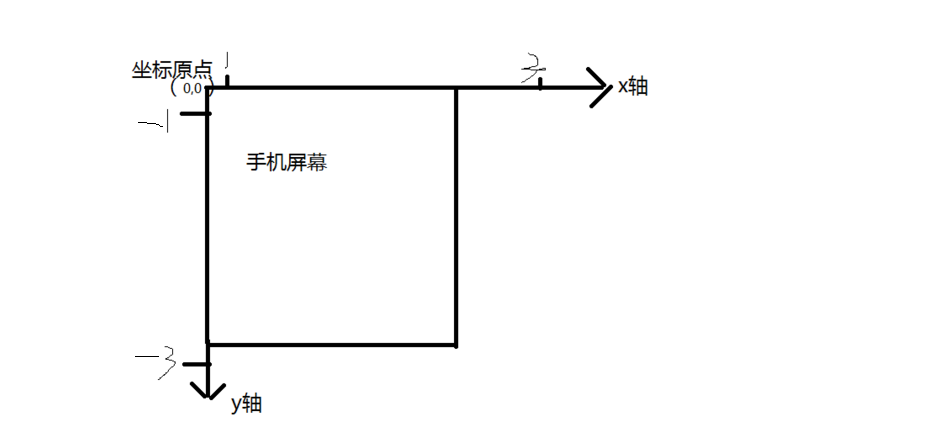
理解下下面的图：

1、屏幕的坐标图，向上滑动屏幕就是x轴不变，y轴由大到小的变化过程：图来解释，这个时候屏幕坐标肯定是比原来屏幕1,3坐标小，因为屏幕在往上滑

2、屏幕的坐标图，向下滑动屏幕就是x轴不变，y轴由小到大的变化过程

3、屏幕的坐标图，向左滑动屏幕就是y轴不变，x轴由大到小的变化过程

4、屏幕的坐标图，向右滑动屏幕就是y轴不变，x轴由小到大的变化过程



图的理解：当前屏幕的参数跟着当前滑动屏幕的参数走，在屏幕滑动时在屏幕上下选个参考物做讲解

来区分上下滑

y1小，y大

x1>x

源码如下：

#defgetSize():获取机器屏幕大小

#defswipeup(t):屏幕向上滑动，x1为x轴坐标，y1起始的y轴坐标，y2结束的y轴坐标

# x1 =int(l[0] \*0.5)：l是上面函数中返回的屏幕大小，0和1其实是索引位，0代表x值，1代表y的值

defgetSize():

x = driver.get\_window\_size()['width']

y = driver.get\_window\_size()['height']

return(x, y)

#屏幕向上滑动（代码里把注释去掉）

defswipeup(t):

l = getSize()

x1 =int(l[0] \*0.5)

y1 =int(l[1] \*0.75)

y2 =int(l[1] \*0.25)

driver.swipe(x1, y1, x1, y2, t)

sleep(6)

swipeup(1000)

#屏幕向下滑动

defswipedown(t):

l = getSize()

x1 =int(l[0] \*0.5)

y1 =int(l[1] \*0.25)

y2 =int(l[1] \*0.75)

driver.swipe(x1, y1, x1, y2, t)

sleep(6)

swipedown(1000)

#屏幕向左滑动

def swipLeft(t):

    l=getSize()

    x1=int(l[0]\*0.75)

    y1=int(l[1]\*0.5)

    x2=int(l[0]\*0.25)

    driver.swipe(x1,y1,x2,y1,t)

#屏幕向右滑动

def swipRight(t):

    l=getSize()

    x1=int(l[0]\*0.25)

    y1=int(l[1]\*0.5)

    x2=int(l[0]\*0.75)

    driver.swipe(x1,y1,x2,y1,t)

这个t是时间

**4.adb指令**

1.adb devices：查看连接得设备

2.adb install apk安装软件

3.adb shell pm list packages查询包名

4.adb unstall 包名字 卸载包

5.adb logcat >路径名 输出日志

6.adb rebot，重启命令

7.adb version：查看adb工具版本

8.adb shell：进入shell模式：shell模式里的命令和linux里面差不多，安卓系统是根据linux开发的

9.

exit：退出shell模式。

**5.app日志怎么看？**

先用adb logcat –c清空，然后adb logcat –v time > ./logcat.log，复现一下崩溃或无响应的场景，再用adb logcat去查看日志即可。

**6.测试app和web端的区别**

相同点：单纯从功能测试的层面上来看的话，APP测试、web测试在流程的功能测试上是没有区别的。

不同点：

**（1）系统结构方面**

Web项目，b/s架构，基于浏览器，web测试只要更新了服务端，客户端就会同步更新。

APP项目，c/s架构，必须要有客户端；APP修改了服务端。则客户端用户所有核心版本都要进行回归测试。

**（2）性能方面**

Web项目：响应时间、cpu等

APP项目：响应时间、cpu，还需要检测流量、电量等

**（3）兼容方面**

Web项目：1.浏览器（谷歌、火狐、IE等）

2.操作系统（Windows、Linux等）

APP项目：1.设备系统iOS、Android、Windows，不同的系统还有不同的系统版本

2.手机型号、屏幕分辨率

**（4）APP有专项测试**

1.中断测试：来电、短信、闹钟、关机、重启等

2.弱网测试：2G、3G、4G、5G、WiFi之间切换

3.耗电量、耗流量、耗CPU

3.安装、更新、卸载

4. 界面操作：横竖屏、手势、多点触控

**（5）测试工具**

Web项目：自动化工具用selenium，性能测试工具用loadrunner、jmeter。

APP项目：自动化工具用appium，性能测试工具用jmeter。

**7.能不能告诉我安卓和ios最近的版本**

(1)Android最近版本

（1）Android最近版本

Android9.0Pie：占比达到了10.4%；

Android8.0/8.1Oreo：增长6.8%，总占比至28.3%；

Android7.0/7.1Nougat：下滑9.9%，总占比19.2%；

（2）IOS最近版本

Ios 13.3 beta

Ios 13.2 beta

Ios 13.1 beta

**8.你用过哪些（安卓）型号测试**

**9.app出现anr（程序无响应）的原因是什么？**

正常来说有两种原因，

1.主线程执行了耗时操作，比如数据库操作或网络编程

2.其他进程（就是其他程序）占用CPU导致本进程得不到CPU时间片，比如其他进程的频繁读写操作可能会导致这个问题。

细分的话，导致ANR的原因有如下几点：

1.耗时的网络访问

2.大量的数据读写

3.数据库操作

4.硬件操作

5.调用thread的join()方法、sleep()方法、wait()方法或者等待线程锁的时候

6.service binder的数量达到上限

7.system server中发生WatchDog ANR

8.service忙导致超时无响应

9.其他线程持有锁，导致主线程等待超时

10.其它线程终止或崩溃导致主线程一直等待。

**10.APP出现crash（崩溃）的原因有哪些？**

app出现崩溃问题，一般就是因为：内存管理错误，程序逻辑错误，设备兼容，网络因素等

比如说

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跑不起来导致崩溃

2.内存泄漏：程序运行时间越长，所占用的内存越大，最终用尽全部内存，导致整个系统崩溃。亦或是非授权的内存位置的使用也会导致app崩溃

3.程序逻辑错误：数组越界，堆栈溢出，并发操作，逻辑错误

4.兼容问题

5.网络因素：因为网络缓慢，无法达到app所需的快速响应时间，导致app 崩溃。或者是不同网络的切换也可能会影响app稳定性

**11.在公司怎么安装你要测的app？**

公司会给我们一个二维码，我们去扫码下载就行了。也有直接给apk包的

**12.能不能讲一下APP网络测试（弱网测试）怎么实现的（弱网测试环境搭建）**

1.fiddler

fiddler的话，首先点击选择栏Rules ---- performance ---- Simulate Modem Speeds

选中后就是开启了网络模拟（默认的上行速度是300，下行速度是150）即

1kb/300ms

如果想修改的话到fiddler的规则中，找到 if （m\_SimulateModem）

去修改上下行的速度即可

**13.appium的原理是什么？**

Appium相当于一个代理服务器，我们从PC上发送请求，中间经过appium，然后appium会转发请求，并转化成移动设备可以识别的命令，接下来移动端会返回执行结果，经过appium，最后将执行结果返回给我们客户端。

Appium 是一个移动端自动化测试开源工具，支持iOS 和Android 平台，支持Python、Java 等语言，即同一套Java 或Python 脚本可以同时运行在iOS 和Android平台，Appium 是一个C/S 架构，核心是一个 Web 服务器，它提供了一套 REST 的接口。当收到客户端的连接后，就会监听到命令，然后在移动设备上执行这些命令，最后将执行结果放在 HTTP 响应中返还给客户端。

**14.手机APP上的元素怎么定位的？**

我们是用uiautomatorviewer工具定位的。

如何查看app包的名称？

第一种，切换成shell模式，然后到data下去查看

第二种，adb shell pm list packages -3

第三种，aapt d badging + 安装包所在的绝对路径

**15.弱网测试的测试点**

弱网测试的话要考虑三种情况，首先是网络正常的情况、其次要考虑无网络的情况、再就是考虑弱网的情况、最后考虑网络状态转变的情况

* 网络正常的情况下各个模块功能是否正常
* 网络正常的情况下，页面元素/数据显示正常
* 无网络情况下，各功能是否可用
* 无网络情况下，各页面的切换是否正常
* 无网络情况下，数据是否会丢失
* 无网络提示信息是否友好
* 无网络情况下，发送请求时是否会导致闪退、卡死的情况
* 弱网情况下，app是否针对请求做了超时处理
* 网络有延迟，操作app数据同步或者OTA升级的时候，会不会发生ANR( 无响应)、崩溃的错误
* 弱网情况下，app回调还没完成的时候，执行其他交互动作，app是否会出现异常
* 弱网情况下，是否会出现请求堆积的情况
* 弱网情况下，app各个页面显示完不完整
* 如果系统超时，提示信息是否清晰明了
* 弱网情况下，app的响应时间是否在一个合理的时间范围内
* 弱网环境下发送的请求在回复网络以后是否会出现重复提交的情况
* 突然断开网络连接，操作APP各个功能是否正常
* 同步数据过程中，断开网络连接，app是否会出现异常
* 传输数据过程中，断开网络连接，app是否会异常

**16.APP性能怎么测？**

我之前测app性能是用GT（随身调）工具来测的。

首先要选择被测试的APP，然后界面上有一些指标可以选择，

PSS（内存）、Private Dirty（私有内存）、CPU、Net（整个手机产生的流量）、jiffies（记录系统开机以来经过了多少tick，一定时间占用的jiffies，可以反映出此进程的CPU消耗）

然后到参数页面这边，可以选择关注的参数，然后勾选你想查看的测试过程中会生成的数据，点击运行即可

在日志页面，可以保存下各个测试的数据，默认会保存在系统根目录下的gt文件夹中

**17.app测试需要关注的指标有哪些**

手机内存、手机CPU使用情况、APP的耗电量、耗流量、APP卡顿情况、APP启动时间

**18.fiddler如何抓手机端的包**

首先到fiddler 中 ，点击Tools --- options，到connections选项卡中设置一下端口号，勾选相关选项，然后到https选项卡下安装证书。

保证移动端和PC端在同一个网络下，到移动端浏览器里输入PC端的IP+你在fiddler里设置的端口号，去下载并安装证书，然后去设置一下代理，进入到网络设置界面将wifi设置为手动代理，设置相关ip地址和端口号，设置好后返回，就可以抓包了

**19.APP测试遇到的BUG**

**20.什么是activity**

Activity一个应用程序的组件，它提供一个屏幕来与用户交互，以便做一些诸如打电话、发邮件和看地图之类的事情

**十六、外出面试题记录**

**1.你们项目前台/后台是什么框架？**

后台框架：

springcloud或者spring

前端框架：

web：vue

UI：element

小程序：原生

安卓开发语言：

Java

iOS语言：  objective-c，当然在iOS开发过程中，也支持c/c++语言与原生的objective-c混编。

**2.xpath与CSS：**

类似点：

1.xpath与CSS都有web页面定位元素的功能

2.xpath与CSS结构类似

区别：

1.CSS定位语法比Xpath简洁，定位方式更灵活多样

2.CSS不支持根据子元素查找元素

3.使用Xpath能够屏蔽掉其他页面元素改变

**3.接口测试的痛点和相关知识**

<http://www.51testing.net/study/improve/69830.html>

**4.jmeter 批量执行接口脚本**

<https://www.cnblogs.com/zhangwuxuan/p/12404111.html>

**5.搜索框测试点**

**功能测试** 1.搜索内容为空，验证系统如何处理 2.搜索内容为空格，查看系统如何处理 3.边界值验证：在允许的字符串长度内外，验证系统的处理 4.超长字符串输入，系统是否会截取允许的长度来检验结果 5.合法的字符串长度后，加空格验证检索结果 6.多个关键字中间加入空格，逗号，tab验证系统的结果是否正确 7.验证每种合法的输入，结果是否正确 8.是否支持检索内容的复制、粘贴、编辑等操作 9.是否支持回车键搜索 10.多次输入相同的内容，查看系统的检索结果是否一致 11.特殊字符、转义字符、html脚本等需要做处理 12.敏感词汇，提示用户无权限等 13.输入的内容是否支持快捷键操作等 14.只能输入允许的字符串长度等 15.输入链接是否正确跳转， 16.搜索的历史纪录是否显示在下面 17.搜索内容有没有联想功能 18.是否可以输入数字，英文，中文 19.是否可以混合输入数字英文中文 20.输入拼音也可以进行检索 21.语音搜索的内容是否匹配 22.断网时，无法搜索 23.进行图片搜索时可以选择拍照或从相册中选取图片进行搜索 24.如果从相册中选取图片进行搜索，图上的大小是否有限制，最大为多少 25.搜索框边上有相机图片，便于图片搜索 26.点击清空历史记录，搜索框是否会清空历史记 2**7.能否**识别图片中的内容 28.点击搜索，显示搜索界面   界面测试 1.查看UI是否显示正确，布局是否合理 2.是否有错别字 3.搜索结果显示的布局是否美观 4.已查看的结果链接，链接的颜色要灰化处理， **5.结果**数量庞大时，页面的分页布局是否合理 6.界面的颜色搭配是否合理 安全性测试 1.脚本的禁用 2.SQL的注入，检索SQL SELECT语句等 3.敏感内容的检索是禁止的 4**.特殊字符**的检索 5.被删除、加密、授权的数据，不允许被查出来，6.是否有安全设计控制 兼容性测试 1.多平台W**indo**ws，mac 2.移动平台android，ios 3.多浏览器火狐、chrome、IE等 性能测试 1.搜索页面的链接打开速度**的时间** 2.搜索出结果消耗时间 3.弱网时搜索的响应时间 4.不同网速下搜索时的响应时间3g,4g,WIFI 易用性 1.有联想功能 2.搜索内容与搜索结果的匹配程度 3.支持拍照搜索，语音搜索

**6.静态测试和动态测试的区别如下：**

1、测试部分的不同

静态测试是指测试不运行的部分：只是检查和审阅,如规范测试、软件模型测试、文档测试等。动态测试是通常意义上的测试，也就是运行和使用软件。

2、测试方式不同

静态测试，通过评审文档、阅读代码等方式测试软件称为静态测试，通过运行程序测试软件称为动态测试。

3、测试方法不同

静态测试是指不用执行程序的测试，它主要采取方案—代码走查、技术评审、代码审查的方法对软件产品进行测试。动态测试主要通过构造测试实例、执行程序、分析程序的输出结果这三种方法来对软件进行测试。