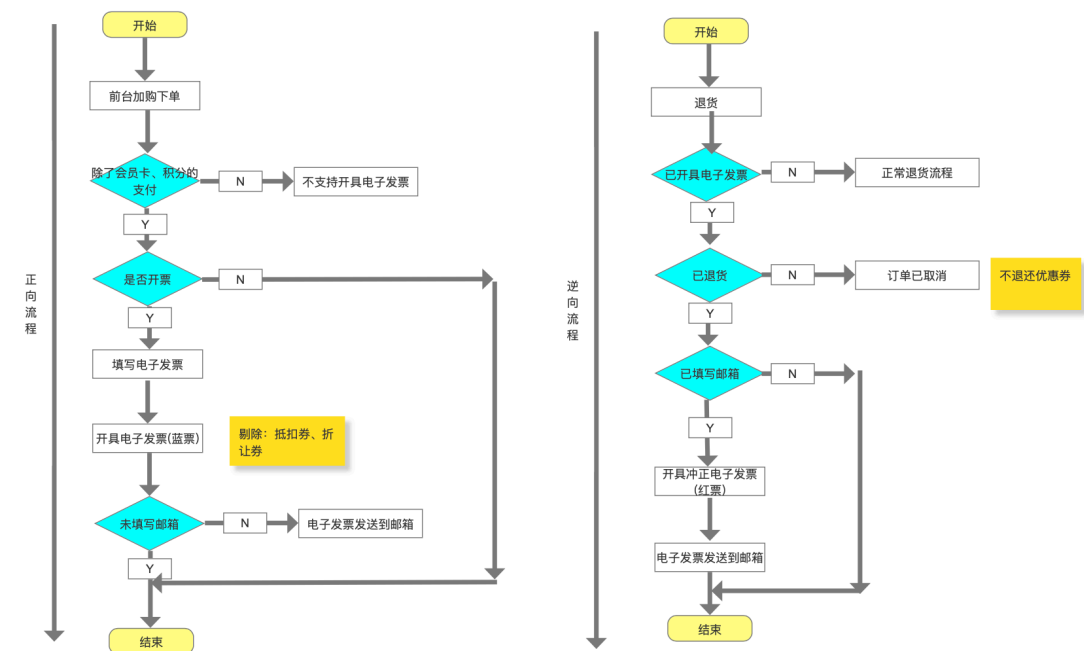


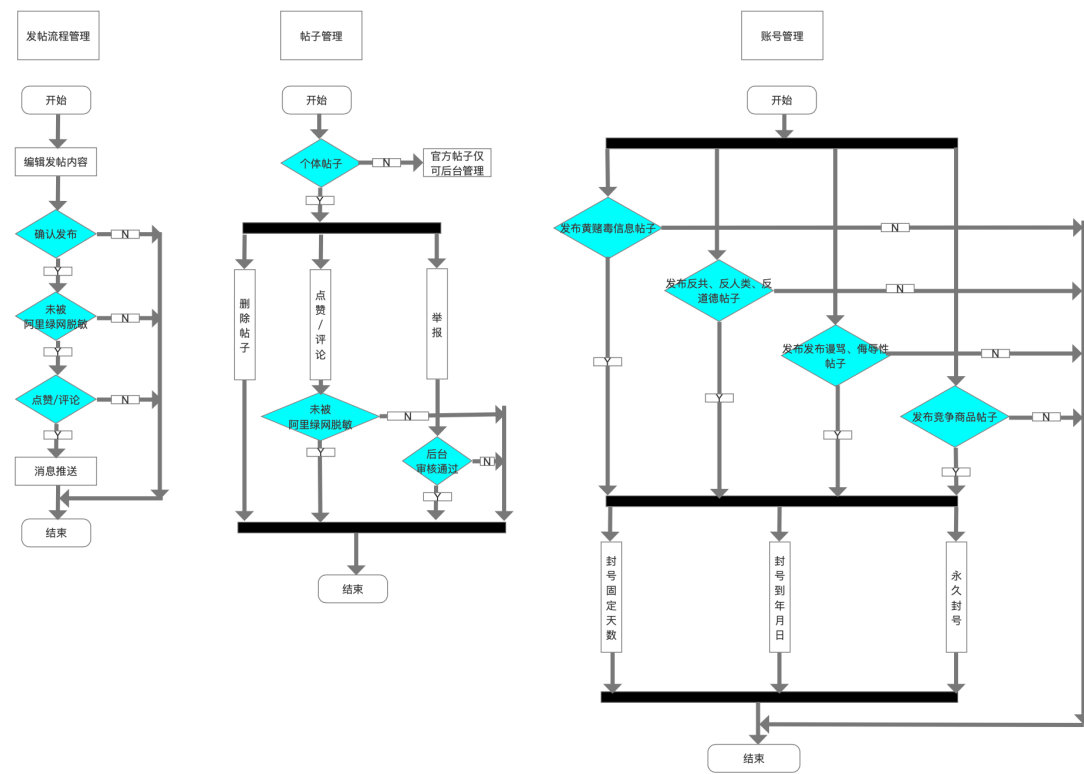
讲述一个测试过程中的业务流程：

电子发票业务流程图：



社区业务流程图：

社区业务流程图



# 西安测试总监提供的自动化测试面试题：

第一阶段：

## 1、 列表与元组的区别

1)元组作为很多内置函数和序列类型方法的返回值存在，也就是说，在使用某些函数或者方法时，它的返回值会元组类型，因此你必须对元组进行处理。

2)元组比列表的访问和处理速度更快，因此，当需要对指定元素进行访问，且不涉及修改元素的操作时，建议使用元组。

3)元组可以在映射（和集合的成员）中当做“键”使用，而列表不行。

其中元祖和列表最大的区别是：列表中的元素可以任意修改，但是元祖中元素不可以修改

## 2、 http的请求流程是怎么样的

1) 大体上可以看作这么几个环节：

TCP建立连接

浏览器发送请求命令

浏览器发送请求头消息

服务器应答

服务器回应头信息

服务器发送数据

断开TCP连接

详细参考：[https://blog.csdn.net/weixin\\_43993151/article/details/96830649](https://blog.csdn.net/weixin_43993151/article/details/96830649)

## 3、 你怎么理解token， cookie， session

详细参考：[https://blog.csdn.net/qq\\_31201781/article/details/94575507](https://blog.csdn.net/qq_31201781/article/details/94575507)

## 4、 怎么获取token，中间会使用到那些思路

1)如果是POSTMAN或者jmeter测试，直接使用postman的js断言，把token参数化或者使用jmeter的正则表达式后置处理其或json字符串的后置处理器截取就可以了，然后其他部分直接引用就可以了

2),如果是在代码里面，使用requests.session方法访问网站，然后使用使用json库去找到token的下标，return或者参数化就可以了

- 5、 如果一个接口存在参数的关联，怎么解决，如何实现的
- a) 如果是实际参数，把有关联的参数剥离出来，放在文档里，使用数据驱动的方式，放在utils的目录下，需要的使用直接调用
  - b) 如果是动态参数，把接口的json格式放在文档里，使用数据驱动和序列化反序列化等技术，通过到动态参数的key值或者下标值，提取到他的参数
  - c) 也可以通过函数之间的调用，把return的参数，传递给需要的函数或者类使用

6、 一个UI的定位存在iframe框架，怎么定位到里面的元素属性

- a) 可以通过 webdriver 提供的 switch\_to\_frame 方法切换表单，然后再定位元素
- b) 也有可能不是表单，而是窗口可以通过 switch\_to\_windows 方法切换窗口再定位

7、 你怎么理解多个元素定位的方式，它的数据类型是什么

- a) findElements()方法可以返回一个符合条件的元素List 组，通过多元素定位可以取出我们需要的元素进行操作

扩展(层级定位): 层级定位的思想是先定位父元素，然后再从父元素中精确定位出其我们需要选取的子元素。层级定位一般的应用场景是无法直接定位到需要选取的元素，但是其父元素比较容易定位，通过定位父元素再遍历其子元素选择需要的目标元素，或者需要定位某个元素下所有的子元素。比较典型的应用是表格的定位。

```
//得到 table 元素对象
WebElement table = driver.findElement(by);
//得到 table 表中所有行对象，并得到所要查询的行对象。
List<WebElement> rows = table.findElements(By.tagName("tr"));
```

8、 在自动化测试中，数据如何分离

- 1. caseID, url, expectedResult 保存在Excel文件中
- 2. request\_data 保存在统一目录的json文件下
- 3. 在utils目录下调用文件进行处理
- 4. 在base目录下进行方法的封装，提供给tests，最后执行测试用例

9、 使用什么模式可以整合web和移动的UI测试

a) 工厂设计模式

第二阶段:

1、 你是怎么把测试技术提高业务测试的覆盖率的

- a) 当先主流的开发模式是前端后端分离,通过接口测试,我们可以区分前后端bug,快速及时的定位到对应开发进行问题处理
- b) 通过弱网测试,可以模拟在网络环境较差的时候出现的问题,比如支付的回调等问题,去定位极端条件下app或者web应用是否异常
- c) 通过UI和API自动化测试,可以提高回归测试的覆盖率,更加快速,高效的定位问题,也可以更加直观的拿出测试凭证,让开发去修复bug,保证回归测试的产品质量,也可以让测试的产出更加直观化
- d) 通过对Linux技术的掌握,可以快速进行日志读取,提高bug定位的效率
- e) 通过对通信知识的扩展,可以剖析应用的B/S架构或者C/S架构在数据交互过程中那个环节出现了问题,更加精准的定位这个问题是bug,还是网络原因,还是请求方式等导致的
- f) 通过对中间件,服务部署等运维知识和对开发技术的加深,可以扩展测试的横向能力,让测试对整个系统认识更加清晰,可以参与到架构讨论、服务迁移、应用熔断等技术型会议,预知技术上线给业务带来的风险,也可以开发出适合当前公司业务发展的测试平台和测试工具等等

2、 接口测试框架你是如何设计的

API自动化测试框架

目录包含:

代码执行前需要初始化测试数据

详细参考: <https://www.cnblogs.com/yinjia/p/9503408.html>

base: 对tests方法进行二次封装

bin: 执行文件,执行所有的测试用例

config: 存放(config文件)

data: 存放数据(csv、excel、xml文件)

pages: 对象层

tests: 测试用例

utils: 对文件操作

Readme.md: 对工程项目的解释

### 3、什么是页面对象设计模

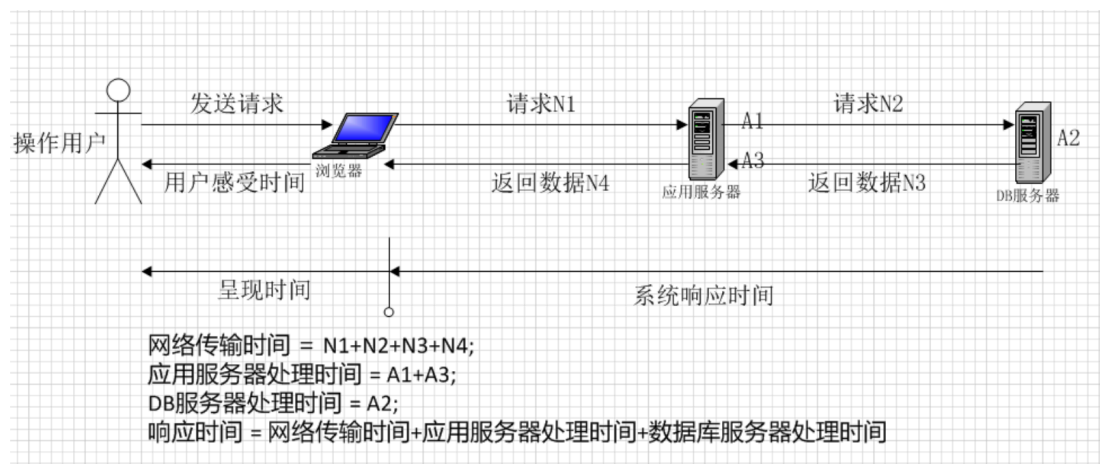
将面向过程转变为面向对象(页面对象), 将测试对象(按钮、输入框、标题等)及单个的测试步骤封装在每个 Page 对象中, 以 page 为单位进行管理。

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, 从而巧妙的避免了当页面元素 id 或者位置变化时, 需要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。可以使代码复用, 降低维护成本, 提高程序可读性和编写效率

- a) base: 对tests方法进行二次封装
- b) bin: 执行文件,执行所有的测试用例
- c) config: 存放 (config文件)
- d) data: 存放数据 (csv、excel、xml文件)
- e) pages: 对象层
- f) tests: 测试用例
- g) utils: 对文件操作

### 4、你怎么理解IOPS, 响应时间

详细参考: <https://www.cnblogs.com/luiheng/p/6710045.html>



### 5、在Linux中如何查看动态的log内容

`tail -f XXX.log | grep 'userid'`

### 6、你的优势是什么

工作认真踏实，勇于挑战，能够处理好和同事、领导之间的关系，熟悉自动化测试，具备牢靠的计算机基础和扎实的编程能力，具备自我学习、自我挖掘、独自解决问题的能力

7、 在之前团队，你是如何定位你的，如果排序，你是第几位，为什么排第一，因为工作过程中积极主动，承担团队中比较重要的项目，同时和同事相处融洽，技术能力较强，所以在团队中经常协助和指导他人工作，同时负责部门内部培训文档的编写和带领新人。

8、 你未来的职业规划是什么，怎么思考的

希望未来的2年，纵向可以加深对自动化测试的理解，横向扩充开发和运维的知识体系，对项目的业务、开发、运维等整体有全局的把控；

个人能够给公司提高测试部门工作效率、提高产品质量，驱动开发规范，促进自动化测试体系完善的同时自己也可以逐渐的往高级自动化测试或者测试开发方向发展

9、 你如何管理团队，和分配任务

参考文档：<https://www.cnblogs.com/dtest/p/4648137.html>

第三阶段：

1、 微服务的优势是什么

服务之间相互独立，打破了传统的单体架构应用模式，解决了业务复杂情况下服务部署和敏捷开发的难题，为devops提供了发展的可能；

多语言生态，打破了开发语言之间的障碍，加快产品开发的进度，最大化团队程序员的优势

- 1、易于维护，可以更快的开始开发
- 2、易于拓展，轻松的对某个组建进行拓展
- 3、技术多样性，技术选型比较自由
- 4、故障隔离，某一个微服务不应该导致整个系统停机
- 5、更高的组织支持，适合小型团队
- 6、独立部署，在部署的时候而不影响其他系统的微服务

2、 它的通信是什么

a、客户端到服务端通信，API Gateway方法。

**API Gateway**是解决微服务通信的一个不错的方法。以客户端为例。一个客户端可以向多个微服务中的任意一个微服务发出请求。**API Gateway**负责请求转发、合成和协议转换。所有请求都要先经过**API Gateway**，然后再将请求转发到对应的微服务中。

#### b、进程通信IPC

这种通信不但可以实现一对一、一对多。还可以实现同步和异步请求。

## 2、 你怎么理解同步通信和异步通信

同步通信一对一：

1. 客户端请求服务端，服务端响应回复客户端请求/响应，期间可能存在超时，网络异常，进程队列等待等，会导致进程堵塞
2. 强耦合，可能会导致redis雪崩，需要hystrix雪崩容错框架支持
3. 单向请求

异步通信一对多：

1. 客户端发送异步的请求，客户端不知道服务端的存在，消息端收到客户端的请求后，最终会被服务端感兴趣的服务器回应
2. 不通过服务端的，发布&&订阅。使用kafka和zookeeper集群，实现生产者和消费者进行异步通信

详细参考：

<https://blog.csdn.net/burpee/article/details/80017246>

## 4、 如果你要监听MQ，你的思路是什么

参考：[https://blog.csdn.net/qg\\_36514588/article/details/82897672](https://blog.csdn.net/qg_36514588/article/details/82897672)

## 5、 你怎么理解契约测试

契约测试的目的是验证服务之间的交互，和是否遵守了服务之间的约束。在契约测试中，需要了解测试消费者与测试生产者，确保生产者与消费者在API是一致的，也就是说验证客户端与服务端HTTP的请求和HTTP的响应，主要应用于测试微服务的API网关和对应的微服务。契约测试会使用到Spring Cloud Contract和pact

契约测试= mock + 接口测试(CI)

其他参考：<https://zhuanlan.zhihu.com/p/58400043>



## 6、你对mock理解多少

参考：

[https://www.cnblogs.com/chaogex/p/3388386.html?utm\\_source=tuicool&utm\\_medium=referral](https://www.cnblogs.com/chaogex/p/3388386.html?utm_source=tuicool&utm_medium=referral)

<https://www.cnblogs.com/sunada2005/p/7727585.html>

## 7、你怎么理解 API 的分层测试(重点)

参考资料：<https://www.jianshu.com/p/14ca0ac451f4>

## HTTP 面试题

### 1，浏览器输入 1 url 按回车背后经历了哪些？

1. 在 PC 浏览器的地址栏输入一串 URL，然后按 Enter 键这个页面渲染出来，这

个过程中都发生了什么事？

1、首先，在浏览器地址栏中输入 url，先解析 url，检测 url 地址是否合法

2、浏览器先查看浏览器缓存-系统缓存-路由器缓存，如果缓存中有，会直接在屏幕中显示页面内容。若没有，则跳到第三步操作。

浏览器缓存：浏览器会记录 DNS 一段时间，因此，只是第一个地方解析 DNS 请求；

操作系统缓存：如果在浏览器缓存中不包含这个记录，则会使系统调用操作系统，获取操作系统的记录(保存最近的 DNS 查询缓存)；

路由器缓存：如果上述两个步骤均不能成功获取 DNS 记录，继续搜索路由器缓存；

ISP 缓存：若上述均失败，继续向 ISP 搜索。

3、在发送 http 请求前，需要域名解析(DNS 解析)，解析获取相应的 IP 地址。

4、浏览器向服务器发起 tcp 连接，与浏览器建立 tcp 三次握手。

5、握手成功后，浏览器向服务器发送 http 请求，请求数据包。

6、服务器处理收到的请求，将数据返回至浏览器

7、浏览器收到 HTTP 响应

8、浏览器解码响应，如果响应可以缓存，则存入缓存。

9、浏览器发送请求获取嵌入在 HTML 中的资源(html, css, javascript, 图片，

音乐.....)，对于未知类型，会弹出对话框。

10、浏览器发送异步请求。

11、页面全部渲染结束。



## 2, GET 和 T POST

GET 在浏览器回退时是无害的，而 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。

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

GET 请求在 URL 中传送的参数是有长度限制的，而 POST 没有。

GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。

GET 参数通过 URL 传递，POST 放在 Request body 中。

长的说：

对于 GET 方式的请求，浏览器会把 http header 和 data 一并发送出去，服务器

响应 200（返回数据）；

而对于 POST，浏览器先发送 header，服务器响应 100 continue，浏览器再发送

data，服务器响应 200 ok（返回数据）。

## 3, cookie 机制和 session 机制的区别

cookie 数据保存在客户端，session 数据保存在服务器端；

cookie 可以减轻服务器压力，但是不安全，容易进行 cookie 欺骗；

session 较安全，但占用服务器资源

## 4, HTTP 状态码

1XX 类状态码信息表示：临时的响应。客户端在收到常规响应之前，应准备接收一个或多个 1XX 响应

2XX 类状态码信息表示：服务器成功的接收了客户端请求

3XX 类状态码信息表示：客户端浏览器必须采取更多操作来实现请求。例如，浏览器可能不得不请求服务器上的不同页面，或者通过代理服务器重复该请求

4XX 类状态码信息表示：发生错误，客户端似乎有问题。例如：客户端请求不存在的页面，客户端为提供有效的身份验证信息

5XX 类状态码信息表示：服务器遇到错误而不能完成该请求

应用层/传输层/网络层/数链路层

## 5, http 和 https 区别？

http 和 https 使用的是完全不同的连接方式，用的端口也不一样，前者是 80，

后者是 443

http 的连接很简单，是无状态的；HTTPS 协议是由 SSL+HTTP 协议构建的可进

行加密传输、身份认证的网络协议，比 http 协议安全。

## 6，什么是 DNS？

域名解析服务。将主机名转换为 IP 地址。如将 `http://www.cnblogs.com/主机`

名转换为 IP 地址：211.137.51.78

## 7，什么是 Http 协议无状态协议?怎么解决 Http 协议无状态协议?

(1)、无状态协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需

要前面的信息

(2) 无状态协议解决办法： 通过 1、Cookie 2、通过 Session 会话保存。

## 接口面试题

### 1，你平常做接口测试的过程中发现过哪些 bug?

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

### 2，接口测试的必要性

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）

### 3，平常你是怎么测试接口的？

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

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

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

**接口安全：**

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

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

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

4、**密码安全规则**，密码的复杂程度校验

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

性能测试

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

**接口响应时间**，响应时间太长了，肯定需要优化，一般都是毫秒级别

## 4，WebService 接口是如何测试的（SoapUI 工具）

<https://blog.csdn.net/vikeyyyy/article/details/80339005>

## 5，没有接口文档如何做接口测试

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

## 6，数据依赖

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

## 7，依赖于第三方数据的接口如何进行测试？

这个标准答案是：mock 进行模拟测试

## 8，查看后端日志，xshell 连上服务器，查看日志

首先在 `xftp` 中确定放日志的路径：

【1】首先连接连接到后台 server

首先了解一些 Xshell 的基本命令，像 cd

(1) cd 是进入下一级目录命令

例如：

cd

/weblogic/Oracle/Middleware/Oracle\_Home/user\_projects/domains/core\_domain/nohuplogs

进入的是 nohuplogs 目录下

(2) tail -f + 日志路径名称， 查看日志动态增加

(3) ll 命令和 ls 命令：都是查看下层目录下含有什么

ll 命令：查看的目录按行详细的展示。大家试下就清楚了。

## 9，弱网

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

## 10，如何分析一个 bug 是前端还是后端的？

平常提 bug 的时候，前端开发和后端开发总是扯皮，不承认是对方的 bug

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

请求报文没问题，那就看返回报文，返回的数据不对，那就是后端开发的问题咯

## 接口自动化面试题

### 1. 接口自动化框架介绍：

Python、unittest、pytest、requests、HTMLtestrunner、Base、config、data、page、tests、logs、report、run

### Unittest：

执行单个测试用例的方法（不同包）：testsuit.addtest('方法名')

执行多个用例的方法（同包）：TestSuit(unittest.makeSuit('类名'))

执行一个包或文件下的多个用例：unittest.TestLoader().discover("包名或者文件名")  
Unittest.testtextrunner(verbosity=2).run(suite)

### 2. json 和字典的区别？ -对基础数据类型的考察

json 本质上还是字符串，只是按 key: value 这种键值对的格式来的字符串

### 3. 测试的数据你放在哪？ -数据与脚本分离

**ini 文件**：账号密码，这种管全局的参数，邮箱配置的一些参数

**随机函数**：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

**参数化**：对于一个接口有多组测试的参数，可以参数化，数据放 yml, text, json, excel 都可以

**数据驱动方式**: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

**静态数据**: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

## 4. 参数化 - 数据驱动模式

参数化的思想是代码用例写好了后,不需要改代码,只需维护测试数据就可以了,并且根据不同的测试数据生成多个用例,可以使用 ddt 或者 paramunittest。

## 5. 下个接口请求参数依赖上个接口的返回数据 - 参数关联

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

## 6. 依赖于登录的接口如何处理

Requests 库对 token 和 session 的管理

## 7. 依赖第三方的接口如何处理

这个需要自己去搭建一个 mock 服务,模拟接口返回数据

## 7. 不可逆的操作, 如何处理, 比如删除一个订单这种接口如何测试

平常手工测试造数据,直接在数据库改字段状态。那么自动化也是一样,造数据可以用 python 连数据库了,做增删改查的操作

测试用例前置操作, setUp 做数据准备

后置操作, tearDown 做数据清理

## 8. 接口产生的垃圾数据如何清理

跟上面一样,造数据和数据清理,需用 python 连数据库了,做增删改查的操作

测试用例前置操作, setUp 做数据准备

后置操作, tearDown 做数据清理

## 9. 一个订单的几种状态如何全部测到, 如: 未处理, 处理中, 处理失败, 处理成功

造数据和数据清理,需用 python 连数据库了,做增删改查的操作

测试用例前置操作, setUp 做数据准备

后置操作, tearDown 做数据清理

造数据, 改数据库订单状态

## 9. python 如何连接数据库操作?

导入 pymysql 第三方库、pymysql.connect 链接数据库，定义一个变量操作 cursor 游标，调用 execute 方法执行 sql，如果是查找不需要 commit，增删改都需要 commit，最后关闭游标和数据库

## 11. 其它的就是运行出报告、代码管理（git）、运行策略和持续集成 jenkins

### selenium

#### 1, selenium 中隐藏元素如何定位？

```
# js 点击 hidden 元素
js = 'document.getElementById("baidu").click()'
driver.execute_script(js)
```

#### 2, 如何判断一个页面上元素是否存在？

方法三：结合 WebDriverWait 和 expected\_conditions 判断

```
from selenium import webdriver
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.ui import WebDriverWait
def is_element_exists(driver, locator):
    ...
```

结合 WebDriverWait 和 expected\_conditions 判断元素是否存在，  
每间隔 1 秒判断一次，30s 超时，存在返回 True，不存在返回 False  
:param locator: locator 为元组类型，如("id", "yoyo")  
:return: bool 值，True or False  
...

```
try:
    WebDriverWait(driver, 30,
1).until(EC.presence_of_element_located(locator))
    return True
except:
    return False
if __name__ == '__main__':
    loc1 = ("id", "yoyo") # 元素 1
    print(is_element_exists(driver, loc1))
```

#### 3, 如何提高脚本的稳定性

优化方向：1. 不要右键复制 xpath(十万八千里那种路径，肯定不稳定)，自己写相对路径，多用 id 为节点查找

2. 定位没问题，第二个影响因素那就是等待了，sleep 等待尽量少用（影响执行时间）

driver.implicitly\_wait(30)这个等待也不要，不要以为是全局的就是好事，有些 js 加载失败时候会一直等，并且页面跳转时候也无法识别

3. 定位元素方法重新封装, 结合 WebDriverWait 和 expected\_conditions 判断元

素方法, 自己封装一套定位元素方法

## 4, 如何定位动态元素

子元素定位父元素, 可以通过 xpath 的语法直接定

位: `..//*[@name="heo"]/..` 两个点.. 就是代表父级元素了

## 5, 平常遇到过哪些问题?? 如何解决的

可以把平常遇到的元素定位的一些坑说下, 然后说下为什么没定位到, 比如动态 id、有 iframe、没加等待等因素

## 6, 一个元素明明定位到了, 点击无效 (也没报错),

### 如果解决?

使用 js 点击, selenium 有时候点击元素是会失效

# js 点击

```
js = 'document.getElementById("baidu").click()'
```

```
driver.execute_script(js)
```

## app

### 1. 什么是 activity?

Android 的最重要的四大组件之一

### 2. Activity 生命周期?

运行 暂停 停止 系统回收

### 3. Android 四大组件?

Android 四大基本组件: Activity、BroadcastReceiver 广播接收器、ContentProvider 内容提供者、Service 服务

### 4. app 测试和 web 测试有什么区别?



#### WEB测试和App测试：

	web测试	app测试
性能测试	只需要测试响应时间	响应时间、流量测试和耗电量测试
兼容性测试	兼容浏览器，兼容IE6，IE8，chrome，firefox	兼容的是手机设备，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。使用Testin这样的商业工具也可以做测试
安装测试	无	安装测试
交叉事件测试	无	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
操作类型测试	无	横屏测试，手势测试
网络测试	无	包含弱网和网络切换测试，重点要考虑回退和刷新是否会造成二次提交。弱网络的模拟，据说可以用360wifi实现设置。
从系统架构的层面	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	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。
升级测试	无	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

## 5. android 和 ios 测试区别？

- 1、Android 长按 home 键呼出应用列表和切换应用，然后右滑则终止应用；
- 2、多分辨率测试，Android 端 20 多种，ios 较少；
- 3、机操作系统，Android 较多，ios 较少且不能降级，只能单向升级；新的 ios 系统中的资源库不能完全兼容低版本中的 ios 系统中的应用，低版本 ios 系统中的应用调用了新的资源库，会直接导致闪退（Crash）；
- 4、操作习惯：Android，Back 键是否被重写，测试点击 Back 键后的反馈是否正确；应用数据从内存移动到 SD 卡后能否正常运行等；
- 5、push 测试：Android：点击 home 键，程序后台运行时，此时接收到 push，点击后唤醒应用，此时是否可以正确跳转；ios，点击 home 键关闭程序和屏幕锁屏的情况（红点的显示）；
- 6、安装卸载测试：Android 的下载和安装的平台和工具和渠道比较多，ios 主要有 app store，iTunes 和 testflight 下载；
- 7、升级测试：可以被升级的必要条件：新旧版本具有相同的签名；新旧版本具有相同的包名；有一个标示符区分新旧版本（如版本号），对于 Android 若有内置的应用需检查升级之后内置文件是否匹配（如内置的输入法）

另外：对于测试还需要注意以下几点：

- 1、并发（中断）测试：闹铃弹出框提示，另一个应用的启动、视频音频的播放，来电、用户正在输入等，语音、录音等的播放时强制其他正在播放的要暂停；
- 2、数据来源的测试：输入，选择、复制、语音输入，安装不同输入法输入等；
- 3、push（推送）测试：在开关机、待机状态下执行推送，消息先死及其推送跳转的正确性；应用在开发、未打开状态、应用启动且在后台运行的情况下是 push 显示和跳转否正确；推送消息阅读前后数字的变化是否正确；多条推送的合集的显示和跳转是否正确；
- 4、分享跳转：分享后的文案是否正确；分享后跳转是否正确，显示的消息来源是否正确；
- 5、触屏测试：同时触摸不同的位置或者同时进行不同操作，查看客户端的处理情况，是否会 crash 等

## 6. app 出现 ANR，是什么原因导致的？

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。

## 7. App 出现 crash 原因有哪些？

内存管理错误，程序逻辑错误，设备兼容，网络因素

## 8. app 对于不稳定偶然出现 anr 和 crash 时候你是怎么处理的？

方法一： app 开发保存错误日志到本地

一般 app 开发在 debug 版本，出现 anr 和 crash 的时候会自动把日志保存到本地

实际的 sd 卡上，去对应的 app 目录取出来就可以了

方法二： 实时抓取

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。

尽量重复操作让 bug 复现就可以了

也可以自己开着 logcat，保存日志到电脑本地，参考这篇：

<https://www.cnblogs.com/yoyoketang/p/9101365.html>

```
adb logcat | find "com.sankuai.meituan" >d:\hello.txt
```

方法三： 第三方 sdk 统计工具

一般接入了第三方统计 sdk，比如友盟统计，在友盟的后台会抓到报错的日志

## 9. app 的日志如何抓取？

app 本身的日志，可以用 logcat 抓取，参考这篇：

<https://www.cnblogs.com/yoyoketang/p/9101365.html>

```
adb logcat | find "com.sankuai.meituan" >d:\hello.txt
```

也可以用 ddms 抓取，手机连上电脑，打开 ddms 工具，或者在 Android Studio 开发工具中，打开 DDMS

关于 ddms 更多的功能，参考这篇：

<https://www.cnblogs.com/gaobig/p/5029381.html>

## 10. 你平常会看日志吗，一般会出现哪些异常

### (Exception) ?

NullPointerException - 空指针引用异常

ClassCastException - 类型强制转换异常。

IllegalArgumentException - 传递非法参数异常。

ArithmeticException - 算术运算异常

ArrayStoreException - 向数组中存放与声明类型不兼容对象异常

IndexOutOfBoundsException - 下标越界异常

NegativeArraySizeException - 创建一个大小为负数的数组错误异常

NumberFormatException - 数字格式异常

SecurityException - 安全异常

UnsupportedOperationException - 不支持的操作异常

## 性能

### 1，重要的是如何分析和定位资源利用和瓶颈

## 面试题实际篇

### 1、上一个公司的项目流程，以及测试的流程；

先开产品会讨论中间的问题以及需求的可行性，再开技术会开发写好技术文档以及接口文档，以及评估出各自的所需要的时间需要给出来，然后前后端进行开发，进行联调，写完测试用例组内会进行测试用例评审，然后开测试用例评审会，执行测试用例，有什么问题提禅道，写测试报告，完了我们测试验收，beta 验收，上线验收

### 2、测试用例的设计；

等价类，边界值，场景法，状态迁移法，正交表等

### 3、linux 命令有哪些；

ls 查看目录中的文件

cd /home 进入 '/home' 目录；cd .. 返回上一级目录；cd ../../ 返回上两级目录

mkdir dir1 创建一个叫做 'dir1' 的目录

rmdir dir1 删除一个叫做 'dir1' 的目录（只能删除空目录）

rm -f file1 删除一个叫做 'file1' 的文件，-f 参数，忽略不存在的文件，从不给出提示。

rm -rf /mulu 目录下面文件以及子目录下文件

cp /test1/file1 /test3/file2 如将/test1 目录下的 file1 复制到 /test3 目录，并将文件名改为 file2

mv /test1/file1 /test3/file2 如将/test1 目录下的 file1 移动到 /test3 目录，并将文件名改为 file2

mv \* ../ Linux 当前目录所有文件移动到上一级目录

ps -ef|grep xxx 显示进程 pid

kill 使用 kill 命令来终结进程。先使用 ps 命令找到进程 id，使用 kill

-9 命令，终止进程。

tar -xvf file.tar 解压 tar 包

unzip file.zip 解压 zip

unrar e file.rar 解压 rar

free -m 查看服务器内存使用情况

### ps 查看进程

2. 如何查看所有 java 进程

grep 是搜索关键字

```
ps -ef | grep java
```

-aux 显示所有状态

```
ps -aux | grep java
```

### 查看日志

5. 如何查看测试项目的日志

一般测试的项目里面，有个 logs 的目录文件，会存放日志文件，有个 xxx.out 的文件，可以用 tail -f 动态实时查看后端日志

先 cd 到 logs 目录(里面有 xx.out 文件)

```
tail -f xx.out
```

这时屏幕上会动态实时显示当前的日志，ctr+c 停止

6. 如何查看最近 1000 行日志

```
tail -1000 xx.out
```

### 查看端口被占用

```
netstat -aon|findstr "49157"
```

```
tasklist|findstr "2720"
```

## 4、使用过什么数据库，多表查询怎么使用；

Mysql, SELECT a.name, b.score

FROM student a, grade b

WHERE a.id = b.id

AND kemu = '数学'

ORDER BY score

DESC

主要问题的是

## 5、app 怎么测试；

1、UI 测试

2、功能测试

3、健壮性测试

4、适配

5、稳定性测试

6、性能测试

7、回归测试

8、上线测试

## 6、HTTP 协议怎么理解；

无状态，三次握手，请求报文，响应报文，响应码

## 6、有没有做过自动化，做过哪些；

做过接口自动化，参与编写优化框架，写接口测试用例

## 7、抓包用什么工具，丢包怎么设置；

Chales,fiddler, chales 弱网设置

## 8、接口怎么测，主要怎么测；

通过性测试，参数组合，安全测试（绕过验证，绕过身份验证，密码加密，加密程度），异常测试（必传非必传，参数类型，参数长度）

性能测试（并发和响应）

## 11、性能测试中遇到过什么，并发多少，怎么分析性能；

参考：<https://blog.csdn.net/zhailihua/article/details/73606433>

## 12、TCP 协议有哪些；

HTTP 协议：超文本传输协议，用于普通浏览

HTTPS 协议：安全超文本传输协议，身披 SSL 外衣的 HTTP 协议

FTP 协议：文件传输协议，用于文件传输

POP3 协议：邮局协议，收邮件使用

SMTP 协议：简单邮件传输协议，用来发送电子邮件

Telnet 协议：远程登陆协议，通过一个终端登陆到网络

SSH 协议：安全外壳协议，用于加密安全登陆，替代安全性差的 Telnet 协议

## 13、遇到开发否认的 bug 怎么解决；

参考链接：[https://blog.csdn.net/weixin\\_40388283/article/details/82749163](https://blog.csdn.net/weixin_40388283/article/details/82749163)

## 14、adb 命令怎么使用；

详见：<https://www.cnblogs.com/bravesnail/articles/5850335.html>