

# 第一章 基础

---

## 配置JDK

- 1、安装jdk1.8.0
- 2、计算机-属性-高级系统设置-环境变量-系统变量
- 3、新建JAVA\_HOME ( jdk安装路径 )
- 4、新建CLASSPATH , 变量值为  
( .;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar; )
- 5、修改path , 变量值为  
( C:\ProgramData\Oracle\Java\javapath;%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\;%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin; )

## jdk ( 开发环境 )

查看jdk版本号 : cmd进dos输入java -version

jdk包含jre和jvm

jre是java的运行环境

jvm是java的虚拟机

## dos命令

tab 补全文件名

打开dos : win+r cmd

查看IP地址 : 网络共享中心-详细信息-ipv4 dos-ipconfig-ipv4 dos-ipconfig/all(查看所有信息)

查看网络ping : ping [www.baidu.com](http://www.baidu.com)

进入根目录 : C盘 c : D盘 d :

dir 查看目录下文件夹 ( 显示当前目录下所有文件及文件夹 ) 【DIR】为文件夹 进入文件夹 : cd Music ( 进入当前目录下的该文件夹 )

cd后跟路径跳转至该位置 , 如cd C:\Users\Administrator\AppData ( 绝对路径 )

cd . 进入当前目录

cd .. 返回上一级目录（相对路径）

cd\ 返回当前盘根目录

cls 清屏

mkdir 创建文件夹

echo 新建名为c.txt的文本文档，把hello写入其中echo hello>c.txt

type 读取文件内容

del 删除文件

rd 删除文件夹

move 移动文件

move .\a.txt .\bbb或者move D : \aaa\a.txt D:\aaa\bbb或者move

copy 复制文件

copy D : aaa\e.txt D:\aaa\bbb\ccc

## 进制转换

二进制转十进制： $11101 (B) = 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 29$

八进制转十进制： $6321 (O) = 6 \cdot 8^3 + 3 \cdot 8^2 + 2 \cdot 8^1 + 1 \cdot 8^0$

十六进制转十进制： $F4A1 (H) = F \cdot 16^3 + 4 \cdot 16^2 + A \cdot 16^1 + 1 \cdot 16^0$

十进制转二进制:除以2取余数（从下往上读）

十进制转八进制：除以8取余数

十进制转十六进制：除以16取余数

二进制转八进制：（从右往左截）每三位读一个数（从左往右读）

二进制转十六进制：每四位读一个数

## 软件后缀

\*.exe 可执行程序

\*.txt 文本文档

\*.doc Word文档，需求文档，接口文档，架构文档，详设文档，数据库说明书，测试报告，测试计划，测试总结

\*.xls \*.csv 表格文档，测试用例

\*\*.\*.ppt PPT幻灯片（2003）报告，汇报工作，演讲

\*.docx \*.xlsx \*.pptx（2007及以后）

\*.cmd/\*.bat 批处理文件

\*.sh Linux操作系统可执行文件

\*.json json格式

\*.ink 快捷方式

\*.xml xml文档，存储数据

\*.java Java源文件

\*.py python源文件

\*.html \*.htm \*.php \*.jsp HTML超文本标记语言

\*.js JavaScript源文件

\*.css 处理网页样式

\*.png \*.jpg \*.gif \*.bmp \*.jpeg常见图片格式

\*.zip \*.rar \*.war \*.jar \*.tar 常见压缩包格式

## 杂识

固定个人电脑IP：控制面板\所有控制面板项\网络和共享中心\本地连接\属性\协议版本4

取共享文件：win+r 输入IP

远程控制：win+r 输入mstsc 对方IP，用户名，密码

三大操作系统：win Linux MacOS

Linux操作系统：Redhat，centos，Ubuntu

软件编程语言：C,C++,C#,JAVA,python，JavaScript，PHP，Ruby，R

三大浏览器：IE，火狐，Chrome

1、什么是host（C:\Windows\System32\drivers\etc）

访问一个网站的时候会去host查看是否具有配置该域名，如果在host文件配置了该域名的ID，将不会解析该域名，直接访问在host配置的id，如果在host文件没有找到该域名，将会对该域名进行解析然后进行访问（提高常用网站的访问速度）

2、禅道如何提交bug、包含产品的选择、指派，bug的优先级，bug的优先级，bug的类型

提bug给开发-->开发解决-->测试指回去-->开发重新解决-->测试关闭bug-->测试重新打开bug-->开发解决-->测试关闭bug

3、如何添加产品，添加项目，上传项目的相关文件

4、开发如何解决bug

5、如何关闭bug

6、如何把bug重新打回去

7、如何重新打开bug

8、其他平台（mantis、jira、redmine）如何提bug，提bug的步骤

9、如果网页打不开

- 1、没有网（检查网线，检查网络是否正常）
- 2、浏览器有故障
- 3、网址是否正确
- 4、服务器是否拒绝请求
- 5、检查防火墙是否打开
- 6、检查是否配置了host
- 7、浏览器是否授权使用网络
- 8、浏览器版本与网页不兼容
- 9、网络比较差导致请求超时
- 10、服务器不存在请求的资源
- 11、服务器不支持当前的http协议
- 12、请求中数据丢失
- 13、服务器出现拥堵
- 14、浏览器设置了代理
- 15、本地的资源占用过多
- 16、Nginx出现问题
- 17、Dns出现了问题
- 18、网站证书出现问题

## 版本控制工具

### 1.svn

安装svn 进行汉化 设置语言 进行检出（地址、用户名、密码）可以下拉到我们需要的东西 学会更新、提交

### 2.git

安装git、去网上下载时（不能下载gitblit）进入git网址 点击版本库 点击自己想下拉的项目 复制出页面上的ssh开头的网址 新建或者打开自己的项目存放的文件夹 在文件夹里面右键 点击git bash（会出现git的命令框）接下来是重点 下拉本地原本没有的项目（从无到有的过程）ssh://test001@192.168.2.2:29418/第八期.git 回车 输入密码 点击回车后可以得到，自己想要的项目内容

如果服务端有更新要更新本地的文件 进入到项目文件（注意不是放项目的文件）右键点击git bash 输入 git pull 回车之后可以下拉到更新后的内容

往服务端提交文件 在想要提交的东西复制到自己项目的文件夹以后 先进行一次 git pull操作 git push 然后再输入 git add. git commit -m '提交的原因' git push origin master（分支名称）

## 第二章（讲义）

### 第一节

测试：是根据需求而进行检测的，但是软件测试并不只是根据需求，也进行常识性错误的检查

测试大纲：为了防止检查的遗漏而把每一个检查点用文件记录下来

- 1 | 把每一个测试点都记录下来，防止遗漏

测试用例：把每一项是怎么进行检查的，记录到一个文件中，一般是Excel

- 1 | 我们一般是用Excel编写的，用例ID、用例名称、用例级别、前提条件、测试环境、操作步骤、预期结果、实际结果、是否通过、设计人员

bug单（缺陷报告）：测试中出现的问题，记录到一个文件里，让作者改好，这个文件就是bug单

软件测试的定义：使用人工和自动手段来运行或测试某个系统的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别。

### 软件测试的项目流程：（ ）

首先产品经理提出需求，提出需求之后开发、测试、产品经理会进行评审，然后开发架构师会根据需求文档设计架构设计文档，然后开发再根据架构文档分别分配各自的模块同时产出详细设计文档，然后进行开发。开发人员开发的同时，我们测试会进行测试用例的编写，编写完成之后开发、测试、产品也一样会进行评审。开发完成后，我们根据测试用例执行测试。发现bug就让开发去解决，当所有问题都解决

完毕之后，我们再进行回归测试，然后让运维发布，到预发布环境然后我们再进行回归，然后再发布到生产环境，再看有没有问题，没有问题就说明上线完成。

## 用到的文档范围：

（用户）需求文档、产品模型

（项目经理）项目计划

（开发）架构师：架构文档（概要文档、架构设计文档、架构设计说明书）

研发人员：开发计划、设计文档、需求规格说明书、规格文档（接口文档、数据库说明书）、详设文档

（测试）测试计划、测试大纲（测试点）、测试用例、bug单（缺陷单及缺陷报告）、测试报告、测试总结

## 软件测试的检查内容：

- 1 我们测试的是软件是否与需求文档一致，然后就找bug让开发修改，确保软件不做不必要的事，还要明确系统有哪些风险

- 保证程序与规范说明一致
- 发现软件中的缺陷
- 确保系统合理执行
- 确保软件不做不必要的事
- 明确发布给用户的系统有哪些风险
- 明确在系统失败之前可以让系统正常运行到何种程度

## 软件测试的一般内容：

- 1 制定测试计划
- 2 设计测试用例
- 3 实施软件测试
- 4 提交缺陷单（缺陷报告或bug单）
- 5 测试总结

## 软件测试的根本目的：

- 1 我们测试的目的呢，主要是保证产品满足用户的需求，尽量发现问题改正问题，交给用户一个高质量软件系统，这就是我们测试的目的

- 1 是交给用户的产品符合用户的需求，
- 2 是产品在用户使用之前尽可能多的发现问题，并改正问题，
- 3 最终把一个高质量的软件系统交给用户使用。

## 软件测试人员必备的素质：

- 1 责任心
- 2 团队合作精神
- 3 理解能力，表达能力
- 4 时时保持怀疑的态度，并且有预防缺陷意识
- 5 具有一定的编程经验

## 第二节

### 软件缺陷产生的原因有哪些:

- 1 在我的测试经历中，发现软件出现问题的原因有同事之间交流不足，导致需求理解不完全，然后就是需求文档不完整或者需求经常变动，然后呢还有就是软件太复杂、时间不够之类的问题

- 1 人员与人员的交流不够，交流上有误解或者根本不进行交流
- 2 程序本身有错误
- 3 软件的复杂性
- 4 需求的不断变化
- 5 工期短，任务重，时间压力大
- 6 参与人员的过度自信
- 7 文档的不完善

### bug的定义：

在软件使用过程中的任何问题（或者是导致软件不能符合设计要求或者不满足消费者需求的问题都可以成为bug）

### 缺陷的分类

按照严重级划分

- 1 1.影响测试进度的问题
- 2 2.死机
- 3 3.功能问题
- 4 4.界面问题
- 5 5.优化和建议等

## 按优先级划分

- 1 1.应立即修复的问题
- 2 2.在产品发布前必须修复的问题
- 3 3.如果时间允许应该修复的问题
- 4 4.可以在版本中存在的问题
- 5 严重级高的问题不一定是优先级高的问题，因为有些问题会推后解决，但是影响进度的问题一定是优先级高的问题。

## 寻找bug的经验：

- 1 1.可以将工作中用到的文档（需求说明书）作为识别和判断的标准
- 2 2.通过和开发人员的沟通可以获知软件设计过程的逻辑过程，分析可能存在的潜在问题
- 3 3.对测试软件产品的行业知识的了解及同行业一般用户的操作习惯来发现问题
- 4 4.善于使用手机，学习和分享其他人判断缺陷的方法和经验。

## 如何保证软件中缺陷的重现：

- 1 首先要善于记录操作步骤，发现bug重现规律，是否是硬件失效，还有是否是bug修改后软件出现了新的问题

- 1 1.不要想当然的接受任何假设，要善于记录操作步骤
- 2 2.要善于查找缺陷重现的规律，是否在特定的时刻出现
- 3 3.关注硬件失效问题，硬件可能不按照预定的方式工作
- 4 4.关注软件的失效问题，对于缺陷的修改，可能会产生新的缺陷。

## 随机错误（无法重现的缺陷）的处理方法：

对于无法再现的缺陷，应对这样的缺陷应该进行详细的记录，并尽快提交给开发人员，对于难以再现的缺陷要合理安排时间，不能因小失大。

## 怎么有效的记录bug



- 1 1. 保证重现缺陷
- 2 2. 分析故障，使用最少的步骤来重现缺陷
- 3 3. 包含所有的重现缺陷的必要步骤
- 4 4. 方便阅读，尽量简单，一个缺陷一个报告
- 5 5. 注意自己的语气

## bug单的内容（ ）

- 1 1. 所属产品
- 2 2. 所属模块
- 3 3. 影响的版本
- 4 4. bug类型
- 5 5. bug标题
- 6 6. bug的严重程度
- 7 7. bug的优先级（高、中、低）
- 8 8. 重现步骤包括三个方面：操作步骤、预期结果、实际结果
- 9 9. bug的重现上传对应的文件（如操作时发现的截图）

## 第三节

### bug的处理流程：（ ）

- 1 首先测试提交缺陷报告给开发处理
- 2 然后开发经理分配缺陷报告给对应的开发人员
- 3 然后开发人员处理缺陷报告，处理完毕后
- 4 测试进行返测，返测不通过，直接打回，返测通过，就关闭缺陷报告。
- 5 一般工作中，测试人员直接提交缺陷报告给对应的开发人员

返测：某一个功能的缺陷被开发修改后，测试人员重新测试看这个问题是否存在。

回归测试：是指开发修改完成这个bug之后，不仅要验证这个bug还要看下是否影响到其他的功能，对这个bug相关的功能进行回归测试。一般上线之前要对上线的所有功能以及影响点进行回归测试。

影响点是如何得出的：一般在测试之前开发和测试都会开一个影响点分析会议，开发会详细描述本次更新的影响范围。

## 缺陷报告的分类

按照处理分类

新提交的、已分配的、待确认的、问题未解决、已解决的、待返测的、已验证的、已关闭的、待归档的、已归档的

按处理意见

已修改的、不是问题、无法修复、以后版本解决、保留、重复、无法重现

## 第四节

### 瀑布模型（ ）

计划⇌需求分析⇌设计⇌编码⇌测试⇌运维

定义阶段 | 开发阶段 | 维护阶段

各阶段产生的文档：（ ）

- 1、计划阶段：项目计划，开发计划，测试计划
- 2、需求分析：需求文档（需求说明书），产品模型
- 3、设计阶段：架构文档（架构设计说明书），详设文档，接口文档——都是由开发人员编写
- 4、编码：开发提出的代码文件，系统部署文档
- 5、测试阶段：测试大纲（一般由leader编写），测试计划，测试用例（case），缺陷报告，测试总结

### 螺旋模型

螺旋模型的每一次迭代都包含了以下六个步骤：

- |   |                         |
|---|-------------------------|
| 1 | 1、确定目标可选方案              |
| 2 | 2、识别和解决项目的风险            |
| 3 | 3、评估技术方案和替代解决方案         |
| 4 | 4、开发本次迭代的交付物和验证迭代产出的正确性 |
| 5 | 5、计划下一次迭代、              |
| 6 | 6、提交下一次迭代的步骤和方案         |

螺旋模型又叫迭代，螺旋模型是软件行业的一种表中的开发规范，如果项目没办法一次性完成，则采用分批次完成，每一次迭代都必须按照螺旋方案走。

### 软件测试的生命周期：

- 1 1、制定测试计划
- 2 2、测试设计和研发
- 3 3、实施软件测试
- 4 4、编写测试报告
- 5 5、版本发布
- 6 6、测试总结

## 软件测试的流程：（ ）

当我们拿到需求之后，首先开发、测试、产品经理会一起对需求进行评审，评审通过后，我们会根据需求文档先列出测试大纲，然后再编写测试用例，同时开发人员也会进行代码的开发，当然测试用例编写完成之后会进行组内、组外的评审，评审过后，开发开发完成会提交一个测试版本，我们开始根据测试用例执行测试，测试过程中如果发现bug，就把bug提交给对应的开发，而且开发改完bug后，我们会进行返测，验证通过了就关闭bug，不通过就继续让开发修改，当所有问题都修复完成之后，然后回归测试，之后让运维发布到预发布环境，我们再进行回归测试，然后再发布到生产环境，再进行回归，最后上线没问题了会写一个测试总结。

## 测试计划的内容：（ ）

- 1 1、项目背景
- 2 2、测试目的
- 3 3、测试起止时间
- 4 4、测试参与人员
- 5 5、测试范围
- 6 6、测试环境要求
- 7 7、任务分配
- 8 8、测试里程碑
- 9 9、测试策略

## 测试报告的内容：（ ）

- 1 测试报告一般由测试负责人用邮件在执行测试过程中每天下班时发送，大致包括：项目名称、测试范围、测试执行时间、测试人员、测试中发现的问题、bug清单

测试报告一般由邮件在执行测试过程中每天下班时发送，大致包括：

项目名称，一般是测试负责人发，（发送给测试、开发和产品），并且抄送相关领导，主要包括测试范围，测试人员，测试执行时间（经过了几轮测试，经过了几天的测试），测试过程中发现的问题（致命的多少个，严重的多少个，一般的多少个，微小的多小的，建议多少个，延期处理的多少个），还有附上buglist

## 测试总结的内容：（ ）

- 1 测试总结一般在线上完成后用邮件发送给相关人员，包括测试的项目、背景、目的、测试环境、测试用例、测试分几个阶段、测试覆盖率、bug数量及分布、遗留的问题；
- 2 最后的结论建议，典型bug原因分析。

1、测试总结一般在线上完成之后再由邮件发送

2、测试的项目、背景、目的、测试环境（软件环境和硬件环境），测试用例，测试分几个阶段，测试结果，测试的覆盖率，发现bug的数量及bug分布（多少个bug，严重、一般、微小、致命等），遗留的问题

3、最后的结论建议，典型的bug分析

## 常见的测试策略（ ）

- 1、数据库测试 2、功能测试 3、性能测试 4、压力测试 5、负载测试 6、接口测试 7、界面测试 8、兼容性测试 9、可靠性测试 10、易用性测试 11、配置测试 12、安全测试 13、安装测试
- 14、加密测试 15、交叉测试 16、文档测试

## 敏捷测试敏捷开发

敏捷开发最大的特点是高度迭代，有周期性，并且能够及时、持续地响应客户的频繁反馈。

敏捷测试的特点：（ ）

- 1、每日站会 2、极限编程 3、测试驱动开发

敏捷测试与传统测试的区别：

- 1、开发测试同时进行
- 2、工作任务划分比较清晰
- 3、耗时或者比较难以解决的问题会在以后的版本中解决
- 4、测试贯穿于整个软件的生命周期

## 第五节

### 测试环境的要求

- 1 1、符合软件运行的最低需求
- 2 2、营造响度简单独立的测试环境
- 3 3、无毒的测试环境

# 软件测试的模型（ ）

1	需求-----（验证与确认）-----验收测试
2	规格说明-----系统测试
3	概要设计-----集成测试
4	详细设计--单元测试
5	编码
6	
7	（软件测试V模型）

1	需求分析	需求测试	
2			
3	概要设计	概要设计测试	
4			
5	详细设计	详细设计测试	
6			
7	编码实现	单元测试	（软件测试的W模型）
8			
9	模块集成	集成测试	
10			
11	系统构建	系统测试	
12			
13	系统安装	验收测试	

## 软件测试的过程

### 单元测试

主要是测试程序代码，为的是确保各单元模块正常。有具体到模块的测试，也有具体到类、函数的测试等。——一般是有开发在开发阶段完成

### 集成测试（联调测试）

在单元测试的基础上，将各单元组成完整的体系，测试软件单位之间的接口是否正确，数据能否正常传递。——比如注册和充值这两个功能能否连通。

### 系统测试（ST测试）

是将一个经过测试的子系统装配成一个完整的系统来测试。它是检验体系是否确实能提供需求中指定功能的有效方法。系统测试的目的是对最终软件系统进行全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

### 验收测试（UAT测试）

验收测试是以用户为主的测试，一般情况下开发测试和产品经理都要参加，验收测试的目的是向客户证明产品是可靠的，符合需求的，必须有用户代表参加（或需求方）

验收测试又分为 $\alpha$ 测试和 $\beta$ 测试

$\alpha$ 测试：是在验收测试的前期，模拟用户的真实环境，全体测试人员和用户代表参与，开发也在现场，发现问题及时解决。

$\beta$ 测试：是在验收测试的后期，在用户环境下进行，有部分测试人员和全体用户参与，开发人员不在现场，发现问题后期解决。

单元测试、集成测试、系统测试之间的区别：就如句子——段落——文章；砖——墙——楼之间的关系

运行维护（运维）是生命周期中持续时间最长的阶段，为了延长软件的使用寿命，适应用户需求，就必须对软件进行维护。包括纠错性维护和改进性维护。

## 测试策略的分类

静态测试：不运行被测程序本身，而是直接通过看代码去寻找程序中可能出现的错误或者评估程序代码的过程。

动态测试：实际运行被测程序本身，输入相应的测试用例，检查预期结果与实际结果的差异。

黑盒测试：又称功能测试，数据驱动测试或基于需求说明书的测试；测试的过程中把被测程序看成一个黑色的盒子，不考虑程序的内部结构，只需要知道该程序输入和输出之间的关系。

白盒测试：又称结构测试，逻辑驱动测试或基于程序本身的测试，测试过程中把被测程序看成一个盒子，白盒测试就是打开盒子，根据程序内容来设计测试用例。

灰盒测试：是介于白盒测试与黑盒测试之间的一种测试，灰盒测试多用于集成测试阶段，不仅关注输出、输入的正确性，同时也关注程序内部的情况。

随机测试：是指测试中所有的输入数据都是随机生成的，其目的是模拟用户的真实操作，并且发现一些边缘性的错误。

常见测白盒测试方法：（ ）

- 1 （1）语句覆盖
- 2 （2）分支覆盖或判断覆盖
- 3 （3）条件覆盖
- 4 （4）判断/条件覆盖
- 5 （5）路劲覆盖

冒烟测试：冒烟测试又叫版本确认测试，主要针对软件的主功能及关键功能的测试。在软件测试前，开发和测试都会对提交测试的版本进行冒烟测试，保证程序的主功能没有问题，以确保测试工作能够正常进行。

手工测试：由测试人员执行测试用例的一种传统测试方法，比较实际结果与预期结果的差异。

自动化测试：是在手工测试的基础上进行测试，运用自动化测试工具执行测试用例。

## 按照测试阶段分类

单元测试-->联调测试-->系统测试-->验收测试

接口测试是测试各个模块互相调用的一种测试。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。测试的重点是要检查数据的交换，以及系统间的相互逻辑依赖关系等。

## 你们公司有几个测试环境

- 1、测试环境（可能有多个环境）
- 2、预发布环境（一个，用生产环境的数据）
- 3、生产环境

## 你们测试几轮

第一轮：全面性的，覆盖性的测试，找出所有问题

第二轮：回归测试，而且我们在预发布环境和生产环境都会进行回归测试。

## 第六节

---

### 软件测试的原则

- 1、尽早的进行软件测试，并且把软件测试贯穿于软件的整个生命周期
- 2、软件测试应该追溯需求
- 3、测试应该由第三方来完成
- 4、不做不充分的测试，也不做过多的测试，因为bug是不可穷举的
- 5、必须彻底检查测试的每个结果
- 6、充分注意bug的集群现象

## 缺陷的二八定理（ ）

在分析、设计、实现阶段的复审和测试工作能够发现和避免80%的缺陷，而系统测试又能找出其余缺陷中的80%，最后的4%的缺陷可能只有在用户大范围、长时间使用后才暴露出来。

## 响应时间的2-5-10原则

就是当用户能够在2秒以内得到响应时，会感觉系统响应很快；

当用户在2-5秒内得到响应时，会感觉系统的响应速度还可以；

当用户在5-10秒内得到响应时，会感觉系统的响应速度很慢，但是还可以接受；

而当用户在超过10秒后仍然无法得到响应时，会感觉系统糟透了，或者认为系统已经失去响应，而选择离开这个web站点，或者发起第二次请求。

## C/S架构与B/S架构（ ）

C/S架构:

需要安装客户端才能够使用的软件。每次更新都需要更新服务端和客户端，比如超市收银系统，每次更新每台电脑都必须重新安装客户端，有分店则更麻烦，人力物力财力都耗费很大。

B/S架构：

只需要一个浏览器就可以访问服务。只需要更新服务端，不需要更新浏览器，用户主动性较高。比如天猫、淘宝、QQ、爱奇艺等。

## 评审的目的：

是否存在不正确的，有冲突的，有歧义或者遗漏的地方

例如：需求文档的评审（开发、测试、产品经理）

测试用例的评审（开发、测试、产品经理）

项目中任何环节产生的文档都是需要评审的

## 怎样正确认识软件测试

- 1 1、软件的质量不是靠测出来的，软件测试只是一种有效提高软件质量的手段
- 2 2、软件测试并不比开发容易
- 3 3、软件测试需要开发人员和测试人员的共同努力
- 4 4、软件测试并不是软件开发后期的一个阶段



## 如果没有需求文档你怎么测试（ ）

如果没有需求文档那么肯定会有提出需求的人，我们首先要找提出需求的人确定好需求点，然后可以拉着开发和产品经理去分析，通过讨论会得出需求的一些分支和细节，开发会讲解他们是如何实现需求，以及测试的时候应该注意的点，然后我们测试会做出测试的思维导图，然后根据思维导图编写测试用例，然后再进行测试。

## 如果测试时间不够怎么办（ ）

首先我们要向测试经理申请看能不能安排同事支援，加急测试

如果没有支援，那么就反馈给项目经理，看能不能多排时间

如果确实不够，我们要首先保证核心业务和主要功能要测一遍，确保正常流程没有问题

然后我们在开发阶段的时候就会督促开发进行自测、单元测试、冒烟测试

以及我们会列出测试点，根据每一个功能点的优先级进行测试，

而且要细化每天的测试任务，加班进行测试，

并且要产出每日的测试报告，发送给项目人员并评估出风险等等

## 上线前发现一个bug（ ）

首先我们要分析bug的严重程度，如果不严重就让开发加班进行修改，然后上线；如果比较严重，先看开发加班是否能修改完成，如果修改不完，就向项目经理申请是否可以延期上线。一般情况下我们会经过几轮测试，在上线前是不会出现此问题的。

## 上线后发现一个bug（ ）

如果在线上（生产环境）发现bug后，我们首先这个bug的严重程度，如果比较严重，我们会回滚到上一个版本，然后进行bug的修改，没有什么问题再进行上线。如果不是很严重，我们会上一个紧急，让开发打一个补丁上去。测试时我们仍然会先发布到测试环境进行测试，再发布到预发布环境，最后再到生产环境。一般情况下我们测试会经过单元、集成、系统、预发布的几轮测试，上线后是不会出现这个问题的。

## 测试时发现一个bug，怎么处理（ ）

发现一个bug，我们首先进行截图保存下来，然后我们提一个正式的bug单给对应的开发人员，然后开发人员去进行处理。

## 发现一个bug，开发认为不是bug怎么办（ ）

首先我们先从需求文档找证据，说法开发，这是一个bug，如果需求文档没有描述清楚或者没有描述，那我们回去找产品经理或者提出需求的人，让他们来定是不是bug。

## 单元测试辅助测试模块

驱动模块：用以模拟待测模块的上级模块。

功能：在集成测试中接受测试数据，把相关的数据传送给待测模块，启动待测模块，并打印出相应的结果。

桩模块：用以模拟待测模块工作过程中所调用的模块。

功能：由待测模块调用，它们一般只进行很少的数据处理，以便于检验待测模块与其下级模块的接口。

## 集成测试的方法

1、增式集成方法：自顶向下集成、自底向上集成

2、非增式集成方法，也叫整体集成方法

## 软件测试人员如何保证版本质量

- 1 1、首先开发提交测试之前都会进行自测和冒烟测试
- 2 2、我们测试过程中会经过集成测试、系统测试、验收测试来保证质量
- 3 3、测试用例我们都会进行评审
- 4 4、上线之前我们一般都会进行交叉测试和回归测试

## 系统发布上线的标准

- 1 1、系统满足了用户所规定的功能和性能的需求
- 2 2、关闭且修复了所有的bug
- 3 3、通过了用户的验收测试
- 4 4、执行了所有的测试用例

## 第三章（数据库）

### url 网址

[https://vip.open.163.com/courses/1840?p=ws\\_11zi](https://vip.open.163.com/courses/1840?p=ws_11zi)

协议://域名（ip）/路径?参数1=值1&参数2=值2&参数3=值3

第一个//到第一个/之间是域名（ip） 第一个/到？之间是路径 问号后面是参数，参数和参数之间用&隔开

http协议默认端口是80 https协议默认端口是443

[www.baidu.com](http://www.baidu.com) 域名

192.168.2.110 IP

## 统一资源定位符

URL里面的空格可以用+表示，或者用%20 表示

URL里面的分隔目录的/ 可以用%2F表示

URL里面的？可以用 %3F 去表示

## plsql的安装步骤

- 1、安装 instant client（指定自己想要的路径）
- 2、安装 plsql 路径（指定一个自己想要的路径）
- 3、打开plsql-tools-preferences.. 添加路径 路径就是instant client的安装路径
- 4、oracle home : D:\Oracle\Instant Client oci libraryu : D:\Oracle\Instant Client\bin\oci.dll
- 5、测试登录（查看是否能够登陆成功）
- 6、为方便登录tools-preferences..-logon里面添加登录历史 plsql连接的是oracle数据库

## Navicat的安装步骤

- 1、解压到自己想要的目录
- 2、目录不能有空格、汉字
- 3、发送Navicat.exe快捷方式到桌面 Navicat连接的数据库是mysql和sqlserver

## mysql连接步骤

- 1、打开Navicat
- 2、左上角连接
- 3、选择数据库类型

4、填写连接信息（连接名（都行），IP（公司的MySQL服务器IP），端口号（默认3306），用户名，密码，连接测试或确定）如果连接的是sqlserver，如果不能直接连上需要安装sqlncli.msi

## 数据库操作

Between Avg Sum Min Max Count As Group by Having Select From Where Like  
Limit Order by Desc Asc Distinct

select 想要的字段 from 表名 where 条件（and或or）条件（<,>,<>,,<=,>=,!=,between,in,notin,is,is not,like）

-- 1、单行注释 单行注释

-- 单行

2、/\*多行注释/多行注释多行注释 多行注释多行注释多行注释 多行注释多行注释多行注释多行注释\*/

3、USE test\_g 指定数据库;

4、show TABLES 查询当前数据库有哪些表格;

5、SELECT `name` FROM student; 查询出单个字段

6、SELECT id,age,`name` FROM student; 查询多个字段

7、SELECT \* FROM student; 查询出所有字段

8、SELECT \* FROM student WHERE sex = '男'; 进行条件筛选

9、SELECT \* FROM student WHERE sex = '男' AND home = '许昌'; 并且条件筛选

10、进行或查询

```
1 SELECT * FROM student WHERE home != '许昌' or home != '安阳';
2 SELECT * FROM student WHERE home <> '许昌' or home <> '安阳';
```

11、在数字40 到50之间

```
1 SELECT * FROM student WHERE 50>= age and age>=40;
2 SELECT * FROM student WHERE age BETWEEN 40 and 50;
```

12、筛选出年龄在40到56之间（用between进行筛选）或者家住苏州的男人的所有信息

13、in , not in 的用法

```
1 SELECT * FROM student WHERE home in ('许昌','安阳');
```

## 14、is is not 的用法

```
1 | SELECT * FROM student WHERE name is NULL;
```

## 15、like的两种用法

```
1 | SELECT * FROM student WHERE name LIKE '%孙%';
```

### -- 1、筛选出学生表32岁的男性

```
1 | SELECT * FROM student where age = 32 and sex = '男';
```

### -- 2、筛选出学生表家庭地址中带'阳'学生的姓名

```
1 | SELECT NAME FROM student WHERE home = '&阳&';
```

### -- 3、筛选出学生表所有姓名为空的数据的所有字段

```
1 | SELECT * FROM student WHERE NAME is NULL;
```

### -- 4、筛选出学生表名字中带'美丽'的人的id

```
1 | SELECT id FROM student WHERE NAME LIKE '&美丽&';
```

### -- 5、筛选出学科表中'曹操'都学了那些学科

```
1 | SELECT SUBJECT FROM SUBJECT WHERE NAME = '曹操';
```

### -- 6、筛选出成绩表中曹操的语文成绩

```
1 | SELECT score FROM score WHERE NAME = '曹操' AND SUBJECT = '语文';
```

### -- 7、筛选出成绩表谭凯的英雄联盟的成绩

```
1 | SELECT score FROM score WHERE NAME = '谭凯' AND SUBJECT = '英雄联盟';
```

### -- 8、筛选出学生表年龄不等于45岁的人的所有信息（两种方法）

```
1 | SELECT * FROM student WHERE age != 45;  
2 | SELECT *FROM student WHERE age>45 or age < 45;
```

-- 9、筛选出学生表年龄大于40小于60的人的名字

```
1 | SELECT NAME FROM student WHERE age BETWEEN 40 AND 60;
```

#1、查询出曹操的平均成绩

```
1 | SELECT avg(score) FROM score WHERE name = '曹操';
```

#2、查询出曹操的总分

```
1 | SELECT SUM(score) FROM score WHERE name = '曹操';
```

#3、查询出学生表一共有多少学生

```
1 | SELECT COUNT(id) FROM student;
```

#4、查询出一共有多少学科

```
1 | SELECT count(distinct SUBJECT) '学科数' FROM SCORE;
```

#5、查询出学生表名字包含'孙'的第二个人的所有信息

```
1 | SELECT * FROM student WHERE NAME = '%孙%' LIMIT 1,1;
```

#6、查询出取出最高分的成绩以及是什么人的什么成绩

```
1 | SELECT * FROM score ORDER BY score DESC LIMIT 1;
```

#7、查询出语文成绩大于60分以上的人姓名

```
1 | SELECT NAME FROM score WHERE `subject` = '语文' AND score >= 60;
```

优先级 库位置>条件>排序>取行>统计>Having

优先级 from-where-group by-select-having-order by-limit

SELECT 聚合函数 ( 字段 ) , 字段 FROM [表格名称] WHERE [条件] [排序] [取行]

```
1 | select 聚合函数 ( 字段 ) , 字段 from 表名称 where 条件 group by 字段
   | having条件order by 字段 limit 6,6 ;
```

SELECT 想要的字段 from 表A INNER JOIN 表B on A.id = B.id;

-- 创建一个表格student,并插入如上数据

```
1 CREATE TABLE student1024(sno INT(10) primary KEY,sname  
char(10),ss char(3),sage int(3),sdept VARCHAR(30));
```

```
1 INSERT INTO student1024(sno,sname,ss,sage,sdept) VALUES  
(95001,'李勇','男',20,'计算机系'),(95002,'刘晨','女',19,'信息系'),  
(95003,'王名','女',18,'数学系'),(95004,'张立','男',19,'信息系');
```

-- 创建一个表格score,并插入如上表格数据

```
1 CREATE TABLE score1024 (sno int(10) /*primary KEY 因为表格中有重  
复数据，不能定义主键*/,cname char(10),grade INT(10));
```

```
1 insert INTO score1024(sno,cname,grade) VALUES (95001,'数据  
库',92), (95001,'英语',85), (95001,'操作系统',88), (95002,'数据  
库',90), (95002,'操作系统',80), (95003,'英语',87), (95004,'数  
学',45);
```

#添加学生王五，学号为95005，性别是男，年龄是23。

```
1 INSERT INTO student1024(sno,sname,ss,sage) VALUES (95005,'王  
五','男',23);  
2 INSERT INTO student1024 (sno,sname,ss,sage,sdept)VALUES  
(95005,'赵柳','女',22,'数据库');  
3 INSERT INTO student1024 VALUES (95008,'麻瓜','男',36,null);  
4 update student1024 SET sno = 95006 WHERE sname = '赵柳';
```

-- 查询student表中的姓名和年龄情况，分别以姓名，年龄作为别名。

```
1 SELECT sname AS '姓名',sage AS '年龄' FROM student1024;
```

-- 查询student表中的性别情况，要求不出现重复行

```
1 select ss,COUNT(ss) FROM student1024 GROUP BY ss ;
```

-- 查询成绩大于80的课程（不能出现重复行）

```
1 SELECT DISTINCT(cname) FROM score1024 WHERE grade > 80;
```

-- 查询成绩为87或80或45的课程名

```
1 | SELECT cname FROM score1024 WHERE grade = 87 or grade = 80 or  
   grade = 45 ;
```

-- 查询姓名是李勇的英语成绩

```
1 | SELECT st.sname,sc.grade FROM student1024 st INNER JOIN  
   score1024 sc on st.sno = sc.sno where sname = '李勇' and cname  
   = '英语';  
2 | select grade from score1024 where sno = (select sno FROM  
   student1024 WHERE sname = '李勇')and cname = '英语';
```

-- 查询各门课程的被选次数

```
1 | SELECT cname,COUNT(cname) FROM score1024 GROUP BY cname HAVING  
   COUNT(cname);  
2 | SELECT cname,COUNT(cname) FROM score1024 GROUP BY cname ;
```

-- 查询各门课程的最高成绩

```
1 | SELECT cname,MAX(grade) FROM score1024 GROUP BY cname;
```

-- 查询男女生的平均年龄

```
1 | SELECT ss,avg(sage) from student1024 GROUP BY ss;
```

## -- 常用指令

SELECT \* FROM student1024;

SELECT \* FROM score1024;

SELECT \* FROM student1024 WHERE sdept = '';

SELECT \* FROM score1024;

TRUNCATE student1024;

TRUNCATE score1024;

DROP student1024;

DROP score1024;

-- 4、查询出选修四科以上的人的基本信息



```
1 | select * from student1024 INNER JOIN score2048 INNER JOIN  
   | subject2048 ON student2048.id=score2048.Student_id AND  
   | score2048.Subject_id=subject2048.id count (group by  
   | student_id)>4;
```

-- 1、查询出张三的所有成绩（输出姓名、学科、成绩）

```
1 | SELECT name,subject,score FROM student032 st INNER JOIN  
   | score032 sc INNER JOIN subject032 su ON st.id=sc.Student_id  
   | AND sc.Subject_id=su.id WHERE NAME ='张三';
```

-- 2、查询出所有人的总分（输出姓名、成绩）

```
1 | SELECT name,SUM(score) FROM student032 INNER JOIN score032  
   | INNER JOIN subject032 ON student032.id=score032.Student_id AND  
   | score032.Subject_id=subject032.id GROUP BY NAME;
```

-- 3、查询出所有人的平均分（输出姓名、成绩）

```
1 | SELECT name,AVG(score) FROM student032 st INNER JOIN score032  
   | sc INNER JOIN subject032 su ON st.id=sc.Student_id AND  
   | sc.Subject_id=su.id GROUP BY NAME;
```

-- 4、查询出选修四科以上的人的基本信息

```
1 | SELECT * FROM student032 st INNER JOIN score032 sc ON  
   | st.id=sc.Student_id GROUP BY ID HAVING COUNT(SC.Student_id  
   | )>4;
```

-- 5、查询出被选课程的人数及任课教师的名字

```
1 | select NAME '姓名',subject '科目',count(subject)'人数'FROM  
   | teacher032 te JOIN subject032 su on su.id=te.subject_id JOIN  
   | score032 sc on su.id=sc.subject_id GROUP BY subject;
```

-- 修改字符名称及字符格式

```
1 | alter table student change column 篮球巨星阿坤 id int(12);
```

-- 修改列的字符格式

```
1 | alter table student modify column id char(12);
```

-- 删除列

```
1 | ALTER TABLE STUDENT DROP COLUMN 篮球巨星阿坤;
```

--加入列

```
1 | alter table student add column 篮球巨星阿坤 char(12);
```

-- 两个表连到一起

```
1 | select * from student , score;
```

-- 表格重命名

```
1 | alter table student032 rename to dalao;  
2 | ALTER TABLE dalao rename to student032;
```

-- 从表插入新行，其外键值不是主键值阻止插入；

--索引相当于图书的目录，可以根据目录中的页码快速找到所需内容，索引一般不能重复；

BEGIN; SQL; SQL; SQL;

-- 手动修改表

```
1 | select * from student for update;
```

--sqlsever的取行操作、top取前几行

```
1 | select top 5 * from student where id not in (select top 5 id  
    from student);
```

-- 创建表格

```
1 | CREATE TABLE <表格名> (字段 字符格式(长度) primary key, 字段 字符格  
    式(长度) ...)
```

-- 插入数据

```
1 | INSERT into <表格名> (表中有的字段) values (插入的内容);
```

-- 更新数据

```
1 | update <表格名> set 要更改的字段 = 更改后的内容 where 条件;
```

-- 删除数据

```
1 | delete from <表格名> where 条件;
```

-- 删除数据

```
1 | truncate table <表格名>;
```

-- 删除表

```
1 | drop table <表格名>;
```

-- 多表联查

-- oracle

-- 取前几行

```
1 | select * from student where rownum <= 3;  
2 | select * from (select rownum t, student.* from student)m where  
   m.t <= 5 and m.t >= 2;
```

-- 对数据修改后需要commit;

```
1 | update student set sex = '男' where name = '曹操';  
2 | commit;  
3 | insert into student values ('020', '朱元璋', '23', '男', '12313893', '南京');  
4 | commit;  
5 | delete from student where name = '朱元璋';  
6 | commit;  
7 | select * from student where name = '朱元璋';  
8 | commit;
```

-- 手动修改表

```
1 | select * from student for update;
```

-- sqlsever的取行操作、top取前几行

```
1 | select top 5 * from student where id not in (select top 5 id  
   from student);
```

-- mysql --免费、开源 mysql AB 默认端口号 ( 3306 ) 易用性

-- win ( 系统 ) linux ( 系统 )

-- oracle --甲骨文公司（收费）默认端口号（1521）安全性较高 可移植性强（拷贝）

--win（系统）linux（系统）

-- sqlserver

--微软（服务端只能安装在win系统上）默认端口号（1433）（收费）

共同点：基本的sql语句都是一样的 但是有一些差别比如说：取前几行，mysql（limit），oracle（rownum），sqlserver（top）

oracle特有的做修改操作的时候要commit

## 几种常用数据库的比较

目前，商品化的数据库管理系统以关系型数据库为主导产品，技术比较成熟。面向对象的数据库管理系统虽然技术先进，数据库易于开发、维护，但尚未有成熟的产品。国际国内的主导关系型数据库管理系统有Oracle、Sybase、Informix和INGRES。这些产品都支持多平台，如UNIX、VMS、Windows，但支持的程度不一样。IBM的DB2也是成熟的关系型数据库。但是，DB2是内嵌于IBM的AS/400系列机中，只支持OS/400操作系统。

### 1.MySQL

MySQL是最受欢迎的开源SQL数据库管理系统，它由MySQLAB开发、发布和支持。MySQLAB是一家基于MySQL开发人员的商业公司，它是一家使用了一种成功的商业模式来结合开源价值和方法论的第二代开源公司。MySQL是MySQL AB的注册商标。MySQL是一个快速的、多线程、多用户和健壮的SQL数据库服务器。MySQL服务器支持关键任务、重负载生产系统的使用，也可以将它嵌入到一个大配置(mass-deployed)的软件中去。

与其他数据库管理系统相比，MySQL具有以下优势：

- 1 (1)MySQL是一个关系型数据库管理系统。
- 2 (2)MySQL是开源的。
- 3 (3)MySQL服务器是一个快速、可靠和易于使用的数据库服务器。
- 4 (4)MySQL服务器工作在客户/服务器或嵌入系统中。
- 5 (5)有大量的MySQL软件可以使用。

### 2.SQL Server

SQL Server是由微软开发的数据库管理系统，是Web上最流行的用于存储数据的数据库，它已广泛用于电子商务、银行、保险、电力等与数据库有关的行业。目前最新版本是SQLServer2005，它只能在Windows上运行，操作系统的系统稳定性对数据库十分重要。并行实施和共存模型并不成熟，很难处理日益增多的用户数和

数据卷，伸缩性有限。SQL Server 提供了众多的Web和电子商务功能，如对XML和Internet标准的丰富支持，通过Web对数据进行轻松安全的访问，具有强大的、灵活的、基于Web的和安全的应用程序管理等。而且，由于其易操作性及其友好的操作界面，深受广大用户的喜爱。

- 1 SQL Server是由微软开发的数据库管理系统；
- 2 广泛用于电子商务、银行、保险、电力等与数据库有关的行业；
- 3 只能在windows上运行；

## Oracle

提起数据库，第一个想到的公司，一般都会是Oracle(甲骨文)。该公司成立于1977年，最初是一家专门开发数据库的公司。Oracle在数据库领域一直处于领先地位。1984年，首先将关系数据库转到了桌面计算机上。然后，Oracle5率先推出了分布式数据库、客户/服务器结构等崭新的概念。Oracle6首创行锁定模式以及对称多处理计算机的支持.....最新的Oracle8主要增加了对象技术，成为关系—对象数据库系统。目前，Oracle产品覆盖了大、中、小型机等几十种机型，Oracle数据库成为世界上使用最广泛的关系数据系统之一。

oracle数据库产品具有以下优良特性。

- 1 兼容性
- 2 可移植性
- 3 可联结性
- 4 高生产率
- 5 开放性

(1)兼容性 Oracle产品采用标准SQL，并经过美国国家标准技术所(NIST)测试。与IBM SQL/DS、DB2、INGRES、IDMS/R等兼容。

(2)可移植性 Oracle的产品可运行于很宽范围的硬件与操作系统平台上。可以安装在70种以上不同的大、中、小型机上;可在VMS、DOS、UNIX、Windows等多种操作系统下工作。

(3)可联结性 Oracle能与多种通讯网络相连，支持各种协议(TCP/IP、DECnet、LU6.2等)。

(4)高生产率 Oracle产品提供了多种开发工具，能极大地方便用户进行进一步的开发。

(5)开放性 Oracle良好的兼容性、可移植性、可连接性和高生产率使Oracle RDBMS具有良好的开放性。

## 第四章 ( Linux )

- 1、安装xshell
- 2、找公司老大要ip地址、用户名、密码
- 3、链接、左上角进行连接输入IP地址点击用户身份验证输入用户名密码 确认
- 4、输入命令进行连接ssh 用户名@ip地址，比如test001@192.168.2.110 回车然后输入账号密码；
- 5、清理屏幕 Ctrl+L 手动输入输入clear也是清空屏幕
- 6、创建文件夹mkdir 加文件名 mkdir -p 文件夹/文件夹 新建多级文件夹
- 7、新建文件 touch 加文件名
- 8、查看当前文件里面那些文件并显示文件的详细信息 ll
- 9、查看当前文件里面有哪些文件 ls
- 10、查看隐藏文件 ls -a
- 11、进入文件 cd空格加文件夹名  
cd ~ 到主目录  
cd - 返回刚才目录  
cd / 到根目录
- 12、ll看到的文件详情中开头是'd'的是文件夹是'-'是文件
- 13、'..'代表上一级的文件
- 14、查看当前路径 pwd
- 15、查看文件 vi 文件名
- 16、编辑文件 vi进入文件后加点击a进行编辑 进入vi界面set number 显示行号
- 17、保存退出 按esc退出编辑 :wq保存退出
- 18、强制保存退出 esc :wq !
- 19、退出 esc :q
- 20、强制退出 esc :q !
- 21、显示行号 set number
- 22、移动文件（重命名）  
mv原名 新名 ；  
mv 原路径/文件名 新路径/文件名

23、查看文件 cat显示所有内容 more按页显示数据 less按行显示数据 点q退出

24、cat

Cat > 文件名 回车 输入文件要输入的内容 点Ctrl+D保存 ( 新建文件并把文件内容写入 )

cat>文件名之后输入的覆盖之前的内容 cat>>文件名 之后输入的追加在之前内容的后面

cat 文件1 > 文件2 > 新文件 把文件1和文件2的内容拼起来写入到新文件里

cat 文件a的完整路径>文件b的完整路径 ( 文件a的内容覆盖文件b的内容 )

cat 文件a的完整路径>>文件b的完整路径 ( 文件a的内容追加到文件b的内容后面并保存 )

cat 路径/文件名 | grep -E '再|见'

25.paste 文件a 文件b 把文件a和文件b的内容以列的形式展示出来

26.cp 要复制的文件 完整路径/新文件的名字

cp -r 复制文件夹的路径/文件夹 要复制到文件夹的路径 复制文件夹

27.rm -rf 删除文件夹 rm 删除文件

28.sh \*.sh 执行sh文件

1、ps 查看进程 ( progress )

ps -ef 所有进程

ps -ef | grep bash ( grep Unix查找 )

ps -ef | grep 应用名 过滤进程

2、kill -9 应用id 杀死进程

3、find 路径 -name 目标文件名\文件夹名

find 路径 -name 目标文件名 -type f 查找文件

find 路径 -name 目标文件名 -type d 查找文件夹

find /home/test026 -name 'test001'

find 路径 -name '\*.txt'

find 路径 -name '010'

find 路径 -name 'test\*' | xargs grep '内容' 找到相应内容并把相关文件的路径及相关行输出

find 路径 -name 'test\*' | rm -rf 删除以test开头的文件和文件夹

find 路径 -name 'test.log' | xargs -I FILE cp FILE ./test001.log 把某路径下的test.log文件复制并更名为test001.log

4、ifconfig 服务器ip

5、tail的用法

tail -f 日志名 查看动态日志

tail -1000f 查看最后一千行动态日志

tail -n 500 查看最新500行的静态日志 ( near )

tail -n +100 目标文件 查看第100行往后的日志

6、head

head -n 10 文件名 前十行

head -n -100 去掉后一百行显示

7、上传和下载

rz 上传文件

sz 路径/文件名 下载 ( 发送 ) 文件到本地

8、grep的用法

grep '再见' 路径/文件名 筛选出文件包含'再见'的每一行，并显示出来

grep -n '再见' 路径/文件名 筛选出文件包含'再见'的每一行，并和行号一起显示出来

cat 路径/文件名 | grep -E '再|见'

grep '再见' 路径/文件名 --col -C3 筛选出文件包含'再见'的每一行，并以不同的颜色显示出来

9、curl https : // [www.baidu.com](https://www.baidu.com) 查看网络是否连通

10、sed -n '3p' test.log 查看第三行

sed -n '3,10p' test.log 显示三到十行的数据

sed -n '/2019-08-18 00:49:52/,/2019-08-18 00:50:52/p' 文件名 查找该时间段内的数据

sed -i 's/再/见/g' 把再改为见



11、本地默认的ip用127.0.0.1或者localhost

12、查看网路是否连通 curl <https://www.baidu.com>

1、查看端口的使用情况 netstat - nutlp

筛选出想要的端口号的占用情况

netstat - nutlp | grep 端口号

netstat -nutlp | grep 端口号

-t (tcp) 仅显示tcp相关选项

-u (udp)仅显示udp相关选项

-n 拒绝显示别名，能显示数字的全部转化为数字

-l 仅列出在Listen(监听)的服务状态

-p 显示建立相关链接的程序名

2、这里显示不同模式下所占cpu时间百分比，这些不同的cpu时间表示：

us, user：运行(未调整优先级的) 用户进程的CPU时间

sy, system: 运行内核进程的CPU时间

ni, niced：运行已调整优先级的用户进程的CPU时间

wa, IO wait: 用于等待IO完成的CPU时间

hi：处理硬件中断的CPU时间

si: 处理软件中断的CPU时间

st：这个虚拟机被hypervisor偷去的CPU时间（译注：如果当前处于一个hypervisor下的vm，实际上hypervisor也是要消耗一部分CPU处理时间的）。

swap交换分区

492536k total — 交换区总量 ( 492M )

11856k used — 使用的交换区总量 ( 11M )

480680k free — 空闲交换区总量 ( 480M )

202224k cached — 缓冲的交换区总量 ( 202M )

3、vmstat查看静态资源占有率

top查看动态资源占有率

free磁盘占有率

#### 4、tar包的压缩

tar -zcvf test.log.tar.gz test.log 把test.log压缩为test.log.tar.gz

tar包的解压

tar -zxvf 目标压缩包 指定位置解压

tar -zxvf 目标压缩包 -C 目标位置

压缩、解压zip包

zip 包名.zip 目标文件

unzip

#### 5、赋权 r : 4 可读

w : 2 可写

x : 1 可执行

drwxr-xr-x. 4 test026 yunchuang 4096 Aug 18 19:32 python

所有者(u)- 所在组(g) - 其他用户(o) 所有者 组 大小 创建时间 文件名

( 短线为权限位置 , 视为空格 )

chmod 777 文件名 赋权u、g、o所有权限

chmod u=rwx 文件名 给所有者添加权限

chmod u=rwx,g=rwx,o=rwx 文件名 赋权ugo所有权限

chmod u+x 文件名 给所有者添加可执行权限

chmod u+x 文件名 去掉所有者可执行权限

chmod -R 777 文件名 把文件夹下的所有文件夹和文件都赋予相同权限

#### 6、df -h 以kb的形式来查看磁盘占有率

7、diff -y 文件1 文件2 比较两个文件的不同点 ( '|'代表不同 , '<'右边有 , '>'左边有 )

8、在当前用户登录其他用户 su 用户名 输入密码

#### 24、服务器之间进行文件互传 文件互传

scp 目标文件 test026@192.168.2.110 : /home/test026/目标文件夹

- 1、安装jdk
- 2、安装jdk的环境变量
- 3、安装tomcat
- 4、bin 启动和关闭Tomcat的文件
- 5、lib 放Tomcat用的jar包
- 6、conf目录 配置文件 ( ) <Connector port="8080" protocol="HTTP/1.1" 改 HTTP端口号 connectionTimeout="20000" redirectPort="8443" />
- 7、logs 日志文件
- 8、webapps 放应用的地方
- 9、temp 存放临时文件
- 10、work 缓存文件存放 <http://localhost:8080/myapp/index.jsp>

## 搭建测试环境 ( )

Tomcat ( bin启动目录 , config配置 , logs日志 , webapps发布目录 )

bin : 存放启动和关闭 tomcat 的脚本文件

conf : 存放 tomcat 服务器的各种配置文件

lib : 存放 tomcat 服务器的支持jar包

logs : 存放 tomcat 的日志文件

temp : 存放 tomcat 运行时产生的临时文件

webapps : 存放 tomcat 的发布文件

work : 存放 tomcat 的工作目录

我们的环境都是由我们测试人员自己搭建的，公司都有搭建环境文档，我们只需要按照文档去安装各种软件，以及配置相应环境变量。我们先安装jdk，再安装Tomcat和数据库。接下来，我们部署测试环境

首先，我们先把服务给停掉，然后把老的代码进行备份，备份完成后，删除webapps下面的老代码，然后把开发给我们的新代码包，通过xshell上传到Tomcat下面的webapps里，然后进行解压，解压完成后，一定要检查代码放的位置对不对，Tomcat端口号有没有冲突，没什么问题去启动Tomcat，并查看它的启动日志，如果启动日志有报错，就把报错信息发给开发，如果没有报错，我们去地址栏访问这个网址，能正常访问，就说明部署成功了。

- 11、重新部署

- 1、开发会把相应的代码给到 ( war包 )
- 2、把Tomcat服务停止(bin目录下的)
- 3、上传war包到服务器 ( 上传rz , 下载sz )
- 4、备份之前的war包 ( cp )
- 5、删除之前的war包 ( rm -rf )
- 6、把开发给的war包复制到webapps文件夹里 ( cp )
- 7、解压war包 ( jar -xvf 文件名 )
- 8、启动Tomcat ( bin的startup )
- 9、查看日志 ( logs文件夹的catAlina.log ) (tail -f)
- 10、进入浏览器应用的网址查看是否能够正常显示 ( tail -f )

## 11、重新部署

- 1、开发会把相应的代码war包给我们；
- 2、从bin目录下的shutdown.sh把Tomcat服务停止；
- 3、上传war包到服务器；
- 4、备份之前的war包；
- 5、删除之前的war包；
- 6、把开发给的war包复制到webapps文件夹里
- 7、jar -xvf 文件名解压war包
- 8、从bin目录下的startup启动Tomcat
- 9、在logs文件夹的catAlina.log里查看日志
- 10、进入浏览器应用的网址查看是否能够正常显示

## 搭建测试环境

- 1 每个公司的具体情况一般不一样，我们公司是参考部署文档来做的；
- 2 首先是安装jdk、Tomcat、MySQL数据库等，然后进行相应的配置；
- 3 配置nginx：在nginx目录中找到nginx.conf文件，配置服务的地址、端口、最大连接数、tomcat连接等信息；
- 4 配置Tomcat的时候：在conf目录下的server.xml文件里配置地址、最大连接数、编译好的应用程序目录等信息，在web.xml文件里配置匹配请求的servlet。

## 部署测试环境

- 1 首先运行Tomcat的lib目录下的shutdown.sh来停止服务器，将开发给的war包上传到服务器，在webapps文件夹备份并删除原测试包，将新war包复制到该文件夹并解压，在lib目录下的startup.sh启动tomcat服务器，之后去logs文件夹下查看日志有没有报错，然后进浏览器验证服务器是否生效。

每个公司的具体情况不一样，可以先说你是参考部署文档来做的，安装jdk,tomcat,mysql数据库等，然后进行相应的配置比如在Nginx目录中找到nginx.conf文件，配置服务的地址、端口、最大连接数、Tomcat连接等

配置tomcat里的在Tomcat目录中找到server.xml和web.xml文件进行配置，简单来说，server.xml里面配置的也是地址、端口、最大连接数、编译好的应用程序目录等信息；web.xml配置的是什么样的请求匹配什么样的servlet；

2. 然后进行部署linux测试环境，开发发包，你用的啥工具XShell等工具连接的应用服务器，用的FTP等工具来更换测试包，在xshell上面经常使用的linux命令说一些就行了啊，比如KILL -9 杀死TOMCAT应用的进程，用FTP先备份上一版本的测试包，再更换当前的测试包，启动服务就行了啊。 ./startup.sh & tail -f ../logs/catalina.out 开启服务并查看启动日志，看提测包是否报错就行了啊

## 第五章（python）

#1、安装python

#2、配置path

#coding = utf-8 编码格式 #注释

```
1 | # 单行注释
```

选中ctrl / 批量注释

"""多行注释"""

五种标准数据类型

1、数字int() a = 123

2、字符串str() b = 'string'

3、列表list[] c = [132,3214,'string']

4、元组tuple() d = (214,214,'string')

5、字典 ( dictionary ) ary{} e = {"ee": 1334,"eee":'string'} print(type())

6、布尔型true/false ture = 1 / false = 0 c = b > a

#c自动判定为布尔值

加+ 减- 乘\* 除/ 余数% 次方\*\* 商取整//

列表list\_a操作

1、定义列表[,,,,,]

2、添加变量

```
1 a.append () 添加一个
2 a.extend([]) 添加多个
3 a.insert (0,12) 在指定位置0添加12
```

3、删除变量

```
1 del a[2] 删除list_a第二位的变量
2 a.remove(12) 在list_a中删除指定变量12
3 a.pop(-2) 删除-2位置的变量
```

## 添加变量#list

c = [36,5,63,6,8,9,34,45]

```
1 c.append(40)#添加一个值
2 c.extend([34,2,66,778])#在后面添加多个值
3 c.insert(0,18)#在第0位插入数值18
4 c.insert(5,'暴力') # 在5位置添加"暴力"
5 del c[2]#删除第2位的元素
6 c.remove(34)#删除指定的数值
7 c.pop(-2)#删除倒数第二位，第零位一直在列表头部
```

## 切片操作

b = a[5] #取对应位置的变量（第六位）

b = a[2:7] 第二位到第七位

b = a[:5] 取前五位

b = a[2:] 取第二位以后

b = a[::-1] 翻转列表

b = a[1:5:2] 在[1:5]区域从头以2的倍数取值

b = a[2:5] + a[0,2] 切片后拼接

## 字符串操作 "\*"

a = 'hello world' b = 'hello python'\n #换行

c = a[0:6] + b[2:6] #与列表相同

## 元组tuple (\*)

长字符串 print ( " 每天早晨被 帅醒 " ) 循环 while循环  
for循环

## 字典 {key:value}

#字典有同键名时后面会覆盖前面的

```
1 d = {'aa':75,'bb':5365,'cc':[425,727],'dd':'cc'}
2 d['ee'] = 'reg' #字典新增内容
3 d['ee'] = 'dsrdg' #字典修改字段
4 del d['ee'] #删除字段 d.pop('aa')
5 a = d.items() #把字典作为列表输出
6 a = d.keys() #把字典的键做为列表
7 a = d.values() #把字典的值作为列表
```

## 格式化

#当参数是一个字段的时候 ( a% ) 多个字段的时候 ( % ( a,a,a ) )

%s 字符串 %d 格式化整数 %f 格式化浮点型

## 遍历

for i in []#可遍历列表及字典转化的列表

for i,j in [('m','f'),('f','g')] print(i,j)

break #直接跳出循环 continue #跳过该次循环 pass #占位符

```
1 for i in range(1,10):
2     if i == 5
3         break
4         print(i)
```

```
1 for i in range(1,10)
2     if i == 5
3         continue
4     else:
5         pass
6     print(i)
```

```
1 exit()/quit()#终止程序
```

# 类

- 1、为什么要去自定义类和函数
- 2、为什么要去实例化
- 3、为什么要有共有参数，私有参数（什么是共有参数）
- 4、为什么要在函数里调用其他函数
- 5、如何实例化
- 6、如何去设置共有参数，私有参数（共有参数其实就是在类里的全局变量）
- 7、如何在函数里面调用其他类？

## #类的继承

- 1、为什么要进行类的继承（当我们相对一部分代码进行修改的时候，我们又不能修改源代码，因为源代码在被其他程序调用，但是我们又不想重新写这部分代码，所有就有了类的继承）
- 2、如何去继承一个类 子继承父的东西
- 3、继承后子和父的关系

```
class inherit_count(count):  
  
pass
```

## 文件操作

```
'''[a'追加,'w'覆盖,'w+', 'r'读取,'r+']'''
```

```
file = open('./','a')#
```

```
file.write('这顿吃什么呢？\n')#具体往文件写入的方法
```

```
ee = file.read()#读取文件内容的方法
```

```
print(ee)
```

```
file.close()#关闭文件的方法
```

r 用只读的方式打开文件，文件的指针放在文件的开头

r+ 打开文件用于读写。文件的指针放在文件的开头

w 打开文件只用于写入，如果文件存在就覆盖写入，如果文件不存在就新建一个文件并写入



w+ 打开文件用于读写，如果文件存在就覆盖写入，如果文件不存在就新建一个文件并写入

a 打开文件用于追加写入，如果文件存在就追加写入，如果不存在就新建一个文件并写入，指针在文件结尾

#容错，允许程序里存在错误，继续执行后面的程序

```
1 try:
2     a = [1,2,3,4]
3     b = a[5]
4     print('b',b)
5 except:
6     print('出错了')
```

```
1 c = [3,4,5,6]
2 d = c[0]
3 print('d',d)
```

#类里面的魔法函数 \_\_init\_\_()

python的一个优势是有着大量自带和在线的模块(module)资源，可以提供丰富的功能，在使用这些模块的时候，如果每次都去网站找在线文档会过于耗费时间，结果也不一定准确。因此这里介绍下python自带的查看帮助功能，可以在编程时不中断地迅速找到所需模块和函数的使用方法

字典：{ } 键、值成对出现，字典是无序的对象集合，键值对可以进行修改

列表：[ ] 列表是有序的，列表内的数据可以修改 字典中的元素是通过键来存取，而列表是通过偏移进行存取

切片：通过下标的方式来获得某一个数据元素，或是通过制定下标的范围来获得一组序列的元素如 [ ] [ : ] [ : : ]

1、Python中数组，字典，元组的区别

列表是有序的，支持索引、切片、合并、删除等操作且可以被嵌套。列表用的是中括号 [ ]

元组 可以放数字、字符、字符串、元组的内容不能进行修改，但可以进行元组和元组之间的合并。元组用括号标识 ( )

2、切片是什么？通过下标的方式来获得某一个数据元素，或是通过制定下标的范围来获得一组序列的元素如 [ ] [ : ] [ : : ]

3、写一个冒泡排序：

```
1 a = [2,5,7,1,6,3,4]
2 for b in range (len(a)-1):
3     for c in range (len(a)-1-b):
4         if a[c+1]>a[c]:
5             a[c],a[c+1]=a[c+1],a[c]
6 print(a)
```

#### 4、如何将一个字符串进行反串输出

```
1 re = "红鲤鱼与绿鲤鱼与驴"
2 aa = re[::-1]
3 print(aa)
```

#### 5、公司里的开发框架SSM ( spring+springmvc+mybatis )

#### 6、求10的阶层

```
1 a = 1
2 for b in range(1,11):
3     a = a * b
4 print(a)
```

#### 7、Python里的装饰器是什么，有什么作用

Python装饰器就是拓展原来函数功能的一种函数，特殊之处在于它的返回值也是一个函数，这个函数是内嵌“原”函数的函数。

作用：

是在不改变原函数名（或类名）的情况下，给函数增加新的功能。

## 第六章（UI自动化）

### 单元自动化测试：

单元测试框架

- Java：Junit、testing
- python：unittest、pytest
- c#：Nunit

### 接口自动化测试：

- 工具：jmeter、soapUI
- Java：httpunit和httpclient（java里的工具包）

- python : requests ( 第三方库 )
- UI ( user interface ) 自动化测试 : selenium ( python ) 、 QTP、 robot framework
- APP自动化测试 ( ) monkey、monkeyrunner、appium

## 什么样的项目适合做自动化测试：

- 1、任务测试明确，不会频繁变动（需求不频繁）
- 2、比较频繁的回归测试
- 3、软件系统界面比较稳定，变动少
- 4、软件的维护周期比较长（项目周期比较长）
- 5、有大量的重复任务的时候
- 6、项目压力不太大的时候
- 7、具备大量的自动化平台
- 8、测试人员具有较强的编程能力

## 自动化条件（\*）：

- 1、软件需求变动不频繁
- 2、项目周期比较长
- 3、自动化脚本可以重复使用

## UI自动化（python3.6.0）搭建环境：

- 1、安装python（pip，环境勾选）
- 2、安装pycharm
- 3、安装selenium（自动化测试工具） pip install selenium==2.53.6
- 4、安装pymysql（连接MySQL数据库工具） pip install pymysql==0.9.3
- 5、安装pillow（图像处理库） pip install pillow==5.4.1
- 6、安装ddt（实现数据驱动） pip install ddt==1.2.0
- 7、安装xlrd（读取Excel库） pip install xlrd==1.2.0
- 8、将HTMLTestRunner放在python的lib目录下（用来生成自动化测试报告）

## 元素定位方式(八种)

( id,name,class\_name,tag\_name,link\_text,xpath,css )

//a[@id='zentao'] //a[@href='/zentao/'] //\*[@id="zentao"]

CSS里有id的话 #id的值 有class .class的值 属性 [属性="属性值"]

```
1 #查找id
2 driver.find_element_by_id("zentao").click()
3 #输入密码12346
4 driver.find_element_by_name('password').send_keys('123456')
5 #点击登录
6 driver.find_element_by_class_name('us_Submit').click()
7 #通过标签名定位
8 driver.find_element_by_tag_name('input').send_keys('admin')
9 driver.find_element_by_link_text("开源版").click()
```

xpath定位(xpath是xml里的一种语言)

```
1 driver.find_element_by_xpath('/html/body/div/div/div[1]/div/div/div/div[2]/a[1]').click()
```

###先定位a , 然后[]内筛选@属性//\*[@id="zentao"]

```
1 driver.find_element_by_xpath('//a[@id="zentao"]').click()
2 driver.find_element_by_xpath('//a[@href="/zentao/"]').click()
3 driver.find_element_by_css_selector('.su_Submit').click()
```

#CSS有ID的话 #id的值 有class的话 .class的值 都没有用其他属性 [属性='属性值']

```
1 driver.find_element_by_css_selector('#zentao').click()
```

#firepath鼠标取目标

```
1 driver.find_element_by_css_selector('html/body/div[6]/div[1]/form/table/tbody/tr[4]/td[2]/input[3]').click()
```

#窗口最大化

```
1 driver.maximize_window()
```

#输入用户名test

```
1 driver.find_element_by_name('username').send_keys('test')
```

#输入密码12346

```
1 | driver.find_element_by_name('password').send_keys('123456')
```

#点击登录

```
1 | driver.find_element_by_class_name('us_Submit').click()
```

#刷新页面

```
1 | driver.refresh()
```

#点击资金管理

```
1 | driver.find_element_by_partial_link_text('资金管理').click()
```

#关闭当前窗口

```
1 | driver.close()
```

#关闭所有窗口

```
1 | driver.quit()
```

#获取标签文本信息

```
1 | text_id = driver.find_element_by_id('zentao').text
```

#获取标签的属性值

```
1 | target_id =  
    driver.find_element_by_id('zentao').get_attribute('target')
```

#判断一个元素是不是可见(布尔值)

```
1 | judge = driver.find_element_by_id("zentao").is_displayed()
```

## #获取元素尺寸

```
1 | cc = driver.find_element_by_id("zentao").size
```

## #清空输入框的内容

```
1 | driver.find_element_by_name("account").clear()
```

## #强制等待

```
1 | time.sleep(3)
```

#隐式等待（对比强制等待）作用于全局，每次定位时等待20秒，未定位到将报错

```
1 | driver.implicitly_wait(20)
```

## #显示等待

```
1 | from selenium import webdriver
2 | from selenium.webdriver.support.ui import WebDriverWait
3 | from selenium.webdriver.common.by import By
4 | import time
5 | url="http://192.168.2.2:81/"
6 | driver=webdriver.Firefox()
7 | driver.get(url)
8 | abc = (By.ID,"zentao")
9 | '''driver.find_element(*abc).click()'''
10 | dd = WebDriverWait(driver,5,2).until(lambda
    x:x.find_element(*abc))
11 | dd.click()
```

## #匿名函数

```
1 | aa = lambda x,y,z:'对号入座'+x+y+z
2 | bb = aa('隐式等待','显示等待','不等待')
3 | print(bb)
4 | '''对比函数
5 | def aa(x,y,z):
6 |     bb = ('对号入座'+x+y+z)
7 |     return(bb)
8 | bb = aa('隐式等待','显示等待','强制等待')
9 | print(bb)
```

```
1 | .//*[@id='ECS_MEMBERZONE']/a[2]#先找到父节点再定位
2 |
3 | import unittest()
4 | class test(unittest.TestCase)
5 |     def setUp(self):
```

## #窗口最大化

```
1 driver.maximize_window()
```

## #5超时时间 , 0.5

```
1 d = WebDriverWait(driver,5,0.5).until(lambda  
  x:x.find_element(*abc))
```

```
1 def findelement(driver,loc):  
2     ele = WebDriverWait(driver,5,0.5).until(lambda  
  x:x.find_element(*loc))  
3     return ele  
4 findelement(driver,abc).click()
```

## #弹窗操作

```
1 #只有一个accept按钮，直接确定  
2 [@value='提示弹窗按钮']").click()  
3 driver.switch_to.alert.accept()  
4 #点击确认弹窗按钮
```

```
1 driver.find_element_by_xpath("html/body/div[1]/input[3]").click()  
2 alert_text_2 = driver.switch_to.alert.text  
3 print("确认弹窗按钮显示文本信息: ",alert_text_2)  
4 time.sleep(1)  
5 driver.switch_to.alert.accept()#确定  
6 driver.switch_to.alert.dismiss()#取消
```

### #点击输入框弹窗按钮

```
1 driver.find_element_by_xpath("html/body/div[1]/input[1]").click()
```

### #在弹窗输入框输入

```
1 driver.switch_to.alert.send_keys("1")  
2 driver.switch_to.alert.accept()  
3 driver.switch_to.alert.dismiss()
```

### #上传文件

```
1 driver.find_element_by_name("file").send_keys("D:\\这是什么.jpeg")
2 #去除输入框的只读属性
3 js =
4     "document.getElementById('demo1').removeAttribute('readonly')"
5 #执行js代码
6 driver.execute_script(js)
7 driver.find_element_by_id("demo1").send_keys("2019-08-27")
```

## #操作滚动条

```
1 url_2 = "http://192.168.2.110/geeknet/user.php"
2 #设置窗口大小
3 driver.set_window_size(600,600)
4 #将滚动条向右移动100个单位，向下移动400个单位#执行js_2代码
5 js_2 = "window.scrollTo(100,400)"
```

## #窗口截图

```
1 driver.get_screenshot_as_file("D:\\picture\\get_screenshot.jpg")
2 )
```

## #下拉列表处理

```
1 from selenium.webdriver.support.ui import Select
2 select_1 = driver.find_element_by_id("s1Id")
3 time.sleep(1)
4 Select(select_1).select_by_visible_text("苹果")
```

## #数据库操作

```
1 import pymysql
2 user = "test026"
3 pwd = "test026"
4 ip = "192.168.2.2"
5 DB = "test"
6 sql = "select age from student where name = '孙权'"
7 #建立数据库连接
8 conn = pymysql.connect(ip,user,pwd,DB)
9 #获取数据库指针
10 cur = conn.cursor()
11 #执行sql语句
12 cur.execute(sql)
```



```

13 #获取执行结果，结果以元组形式存在
14 num = cur.fetchone()
15 print(num)
16 print("id = %s name = %s age = %s sex = %s tel = %s home = %s"
17       %(num[0],num[1],num[2],num[3],num[4],num[5]))
18 def get_mysql_value(ip,user,pwd,DB,sql):
19     #建立数据库连接
20     conn=pymysql.connect(ip,user,pwd,DB)
21     #获取数据库指针
22     cur=conn.cursor()
23     #执行sql语句
24     cur.execute(sql)
25     #获取执行结果 结果以元组的形式存在
26     num=cur.fetchone()
27     return num[0]
28     #关闭指针
29     cur.close()
30     #关闭数据库连接
31     conn.close()
32 a=get_mysql_value(ip,user,pwd,DB,sql)
33 print(a)
34 #关闭指针
35 cur.close()
36 #关闭数据库连接
37 conn.close()

```

## #区域截图

```

1 #截图并保存
2 driver.get_screenshot_as_file("bos.png")
3 #获取元素的位置
4 loc=driver.find_element_by_id("loginform:vCode").location
5 print(loc)
6 #获取元素的大小
7 kw=driver.find_element_by_id("loginform:vCode").size
8 print(kw)
9 x=loc['x']
10 y=loc['y']
11 xx=loc['x']+kw['width']
12 yy=loc['y']+kw['height']
13 #打开截图
14 a=Image.open('bos.png')
15 #按照坐标截图

```

```
16 im=a.crop((x,y,xx,yy))
17 #保存截图
18 im.save("code.png")
```

#获取cookie cc=driver.get\_cookies()

```
1 from selenium import webdriver
2 import time
3 url="http://192.168.2.110/geeknet/user.php"
4 driver=webdriver.Firefox()
5 driver.get(url)
6 cookie = [{'name': 'ECS_ID', 'value':
    '8c8803beda9fb0c12951a722c42f32d50d4a9676', 'path': '/',
    'domain': '192.168.2.110', 'secure': False, 'httpOnly': True,
    'expiry': None},
7   {'name': 'ECS[visit_times]', 'value': '1', 'path': '/',
    'domain': '192.168.2.110', 'secure': False, 'httpOnly': True,
    'expiry': 1598447018},
8   {'name': 'ECS[username]', 'value': 'test', 'path': '/',
    'domain': '192.168.2.110', 'secure': False, 'httpOnly': True,
    'expiry': 1568207018},
9   {'name': 'ECS[user_id]', 'value': '15306', 'path': '/',
    'domain': '192.168.2.110', 'secure': False, 'httpOnly': True,
    'expiry': 1568207018},
10  {'name': 'ECS[password]', 'value':
    '9c7eb0889e2c5970a15e5442a5aaa8e6', 'path': '/', 'domain':
    '192.168.2.110', 'secure': False, 'httpOnly': True, 'expiry':
    1568207018}]
11 for i in cookie:
12     driver.add_cookie(i)
13 time.sleep(3)
14 driver.refresh()
15 time.sleep(2)
16 driver.quit()
```

## #显示等待

```
1 def findelement(driver,loc):
2     ele = WebDriverWait(driver,5,0.5).until(lambda
    x:x.find_element(*loc))
3     return ele
4
5 from selenium import webdriver
6 import time,random
```

```

7 import unittest
8 from HTMLTestRunner import HTMLTestRunner
9 class Test_Login(unittest.TestCase):
10     def setUp(self):
11         url = "http://192.168.2.110/geeknet/user.php"
12         self.driver = webdriver.Firefox()
13         self.driver.get(url)
14     def test_login1(self):
15         #输入用户名test013
16
17         self.driver.find_element_by_name("username").send_keys("test0
18         13")
19         #输入密码
20
21         self.driver.find_element_by_name("password").send_keys("qq101
22         01100")
23         #勾选保存登录信息
24         self.driver.find_element_by_id("remember").click()
25         #点击登录
26         self.driver.find_element_by_name("submit").click()
27         time.sleep(3)
28
29         a=self.driver.find_element_by_css_selector(".f4_b").text
30         if a=="test013":
31             print("登录成功")
32     def test_login2(self):
33         #输入用户名test013
34
35         self.driver.find_element_by_name("username").send_keys("test"
36         )
37         #输入密码
38
39         self.driver.find_element_by_name("password").send_keys("12345
40         6")
41         #勾选保存登录信息
42         self.driver.find_element_by_id("remember").click()
43         #点击登录
44         self.driver.find_element_by_name("submit").click()
45
46         a=self.driver.find_element_by_css_selector(".f4_b").text
47         print(a)
48         self.assertTrue(a=='test')
49     def tearDown(self):
50         self.driver.close()
51 if __name__ == "__main__":

```

```

42     #建立一个空的单元测试集合
43     suit= unittest.TestSuite()
44     #向测试集合里添加测试用例
45     suit.addTest(Test_Login('test_login2'))
46     suit.addTest(Test_Login('test_login1'))
47     now = time.strftime("%Y-%m-%d %H-%M-%S")
48     filename = "../report/" + now + "result.html"
49     fp = open(filename,"wb")
50     runner = HTMLTestRunner(stream=fp,title="登录测试报
告",description="用例描述")
51     #执行测试集合
52     runner.run(suit)
53     fp.close()

```

## #任何类都继承object

```

1  from selenium import webdriver
2  from selenium.webdriver.support.ui import WebDriverWait
3  from selenium.webdriver.common.by import By
4
5  class Base(object):
6      '''基于selenium进行二次封装'''
7      #初始化函数
8      def __init__(self,driver:webdriver.Firefox):
9          self.driver = driver
10         # cc = (By.ID,"zentao")
11         #loc是参数的定位方式，如 (By.ID,)
12         def findelement(self,loc):
13             print("正在定位元素信息： 定位方式->%s value的值是->%s"%
(loc[0],loc[1]))
14             ele = WebDriverWait(self.driver,5,0.5).until(lambda
x:x.find_element(*loc))
15             return ele
16         def sendkeys(self,loc,text):
17             #向定位的元素输入文本
18             return self.findelement(loc).send_keys(text)
19         def click(self,loc):
20             return self.findelement(loc).click()
21         def clear(self,loc):
22             return self.findelement(loc).clear()
23         def get_text(self,loc):
24             try:
25                 a = self.findelement(loc).text
26                 return a

```

```

27         except:
28             print("获取文本失败，返回为空")
29             return ""
30
31 if __name__ == "__main__":
32     loc_user = (By.NAME, "username")
33     loc_pwd = (By.NAME, "password")
34     loc_sub = (By.CLASS_NAME, "us_Submit")
35     loc_login_user = (By.CSS_SELECTOR, ".f4_b")
36

```

```

1 driver = webdriver.Firefox()
2 driver.get("http://192.168.2.110/geeknet/user.php")
3 dd = Base(driver)
4 dd.sendkeys(loc_user, "test026")
5 dd.sendkeys(loc_pwd, "test026")
6 dd.click(loc_sub)
7 t = dd.get_text(loc_login_user)
8 if t == "":
9     print("pass")

```

```

1 from selenium import webdriver
2 from selenium.webdriver.support.ui import WebDriverWait
3 from selenium.webdriver.common.by import By
4
5 class Base(object):
6     def __init__(self, driver: webdriver.Firefox):
7         self.driver = driver
8     def findelement(self, loc):
9         print("正在定位元素信息：定位方式->%s value的值是->%s"%
10             (loc[0], loc[1]))
11         ele = WebDriverWait(self.driver, 5, 0.5).until(lambda
12             x: x.find_element(*loc))
13         return ele
14     def sendkeys(self, loc, text):
15         return self.findelement(loc).send_keys(text)
16     def click(self, loc):
17         return self.findelement(loc).click()
18     def clear(self, loc):
19         return self.findelement(loc).clear()
20     def get_text(self, loc):
21         try:
22             a = self.findelement(loc).text
23             return a

```

```
22         except:
23             print("获取文本失败，返回为空")
24             return ""
25 if __name__=="__main__":
26     loc_user=(By.NAME,"username")
27     loc_pwd=(By.NAME,"password")
28     loc_sub=(By.CLASS_NAME,"us_Submit")
29     loc_login_user=(By.CSS_SELECTOR,".f4_b")
30     driver=webdriver.Firefox()
31     driver.get("http://192.168.2.110/geeknet/user.php")
32     dd=Base(driver)
33     dd.sendkeys(loc_user,"test032")
34     dd.sendkeys(loc_pwd,"666666")
35     dd.click(loc_sub)
36     t=dd.get_text(loc_login_user)
37     if t=="":
38         print('pass')
```

## UI自动化复习

### 一、手工测试和自动化的优缺点：

测试工作无论是手工测试还是自动化测试都是软件质量保障的一个途径。对于一些基本的逻辑性不强的操作，可以使用自动化测试工具。对于一些逻辑性很强的操作，如果自动化测试不是很健全，不建议使用，这时就需要手工测试了。手工测试也存在着一些缺陷。手工测试最常做的是重复的手工回归测试。不但代价昂贵，而且容易出错。

### 二、自动化测试的优点：

- 1、对程序的回归测试更方便
- 2、可以运行更多更繁琐的测试
- 3、可以执行一些手工测试能进行或者是不可能进行的测试（比如说压力测试和模拟大量用户操作）

### 三、自动化测试的缺点：

- 1、不能取代手工测试
- 2、手工测试比自动化测试发现的缺陷更多
- 3、工具本身并没有想象力

综上所述可以归结为自动化完成不了的，手工测试可以弥补两者有效结合，是测试质量保证的关键。

## 四、会不会搭建自动化测试环境

搭建过自动化测试环境，我去公司之前已经搭建好了，我们公司有一个十年工作经验的自动化测试工程师，是他搭建的自动化框架。公司已经有完整的自动化测试环境搭建手册，我们只需要按照手册一步一步安装需要的软件，以及配置相应的环境变量，比如，我们公司采用的是python+selenium+unittest+ddt框架以及POM设计模式（PO）来进行自动化测试的。我们先安装python，安装selenium，安装pip，安装pillow，ddt，pymysql，HTMLTestRunner等第三方库，安装pycharm，然后配置相应的环境变量就可以了。

### 五、POM的优势：

（POM：page object model 页面对象模型）

- 1、POM提供了一种在ui层操作，业务流程验证分离的模式，这使得测试代码变得更加清晰并且具备高可读性；
- 2、对象库与用例分离，使得我们更好的复用对象，甚至能与不同的工具进行深度的结合与应用，可复用的页面方法代码会变得更加优化。

（①业务流程：把每个页面都封装在一个类里面，把类里面把每一个操作都封装成一个方法。②用case里面的测试用例验证业务流程；对象库就是引用的第三方库，比如读取excel文件的第三方的库、连接数据的库，都要封装在common目录下，而我们的测试用例封装在case里面，数据在data。这些都是分离的，可以更好的调用。可复用的页面：每一个页面都封装成一个类。在调用的时候只需要调用方法就可以了，可以大大减少代码量。）

## 六、一个项目当中，功能测试和自动化测试各占多少比例？

一般自动化测试不会超过百分之30，还是以功能测试为主

自动化测试和功能测试所占比例在3:7左右

## 七、自动化测试是怎样组织测试用例的？/自动化测试时怎样识别脚本中哪些是case？

我们的测试用例都是放在case目录下，通过unittest单元测试框架里的discover方法筛选出以test开头的测试用例，循环添加到测试套件中让系统执行筛选出的全部用例。比如：

```
1 test_dir='./case/'
2 a=unittest.defaultTestLoader.discover(test_dir,pattern="test*.py")
```

## 八、你们测试数据是怎么进入到测试用例里面的？

我们首先通过Xlrd第三方库来封装读取excel文件的方法。通过封装的方法读取放在data目录下的excel数据文件，放在一个列表里，通过data数据驱动框架，将列表中的数据循环添加到测试用例里。

## 九、自动化测试中，下拉框列表是怎么处理的？

### api\_010

首先引入Select方法，然后定位元素，把元素的位置传给select方法，调用Select方法里的一些操作，比如visible\_test和value

```
1 from selenium.webdriver.support.ui import Select
2 select_1 = driver.find_element_by_id("s1Id")
3 Select(select_1).select_by_visible_text("苹果")
4 select_2 = driver.find_element_by_id("s2Id")
5 time.sleep(1)
6 Select(select_2).select_by_value("o2")
```

## 十、自动化怎么进行截图？ api\_009

调用get\_screenshot\_as\_file的方法进行截图，括号里命名保存

```
1 from selenium import webdriver
2 driver.get_screenshot_as_file("D:\\picture\\get_screenshot.jpg")
3 )
```

## 十一、python 如何连接数据库？ api\_011

①引入pymysql第三方库

```
1 import pymysql
```

②准备连接数据库的用户名，密码，IP地址，数据库名称，以及要执行的sql语句



```
1 user = "test026"
2 pwd = "test026"
3 ip = "192.168.2.2"
4 DB = "test"
5 sql = "select age from student where name = '孙权'"
```

### ③建立数据库连接,获取数据库指针,执行sql语句

```
1 #建立数据库连接
2 conn = pymysql.connect(ip,user,pwd,DB)
3 #获取数据库指针
4 cur = conn.cursor()
5 #执行sql语句
6 cur.execute(sql)
```

### ④获取执行结果,最后关闭指针,关闭数据库链接。

```
1 #获取执行结果,结果以元组形式存在
2 num = cur.fetchone()
3 print(num)
4 print("id = %s name = %s age = %s sex = %s tel = %s home = %s"
5       %(num[0],num[1],num[2],num[3],num[4],num[5]))
6 #关闭指针
7 cur.close()
8 #关闭数据库连接
9 conn.close()
```

```
1 #数据库
2 def get_mysql_value(ip,user,pwd,DB,sql):
3     #建立数据库连接
4     conn=pymysql.connect(ip,user,pwd,DB)
5     #获取数据库指针
6     cur=conn.cursor()
7     #执行sql语句
8     cur.execute(sql)
9     #获取执行结果 结果以元组的形式存在
10    num=cur.fetchone()
11    return num[0]
12    #关闭指针
13    cur.close()
14    #关闭数据库连接
15    conn.close()
16 a=get_mysql_value(ip,user,pwd,DB,sql)
17 print(a)
```

## 十二、如何处理时间控件？ api008

运行js代码去除输入框只读属性，随后定位输入框并键入时间，时间的格式需符合控件格式要求

```
1 | js =  
  | "document.getElementById('demo1').removeAttribute('readonly')"  
2 | driver.execute_script(js)
```

## 十三、如何处理弹框：api006

首先我们先确定到弹框的位置，然后从网页页面转换到弹框上并将弹框的内容用文本输出出来。

### ①确认弹框

```
1 | dreiver._switch_to.alert.accept
```

### ②取消弹框

```
1 | dreiver._switch_to.alert.dismiss
```

### ③输入弹框

```
1 | dreiver.switch_to.alert.send_keys("输入内容")
```

## 十四、元素定位方式都有哪些

### 1、ID定位

```
1 | driver.find_element_by_id("account")
```

### 2、Name定位

```
1 | driver.find_element_by_name("account")
```

### 3、link\_text全部文本定位

```
1 | driver.find_element_by_link_text("开源版")
```

### 4、partial\_link\_text部分文本定位

```
1 | driver.find_element_by_partial_link_text("开源")
```

## 5、Class name定位

```
1 | driver.find_element_by_class_name("us_Submit")
```

## 6、tag\_name标签名定位

```
1 | driver.find_element_by_tag_name("input")
```

## 7、Xpath定位

```
1 | driver.find_element_by_xpath("/html/body/div/div/div[1]/div/div/div/div[2]/a[1]")#绝对路径
2 | driver.find_element_by_xpath("//a[@id='zentao']")#相对路径
```

## 8、Css定位

```
1 | driver.find_element_by_css_selector("#zentao")
```

# 十五、如何处理windows自带的上传框：api\_007

### 1、使用Autolt v3小工具

1. 使用Autolt window Info定位
2. 使用SciTe Script Editor编写脚本
3. 使用Compile Script Info将脚本打包为EXE文件

```
1 | ControlFocus("文件上传","", "Edit1")
2 | WinWait("[CLASS:#32770]", "", 10)
3 | ControlSetText("文件上传","", "Edit1", "D:\Tomcat.zip")
4 | Sleep(2000)
5 | ControlClick("文件上传","", "Button1")
```

### 2、在pycharm中运行Autolt工具

定位后点击上传框按钮，然后运行Autolt脚本

```
1 | import os
2 | driver.find_element_by_name("file").click()
3 | os.system("D:\\autoIt.exe")
```

# 十六、如何截取部分图片？ api\_012

首先我们要引入PIL方法里的Image

### 1、打开web页面后进行截图保存（get\_screenshot\_as\_file方法）

- 2、获取目标元素位置大小 ( location , size )
- 3、实例化后获取元素的坐标
- 4、横坐标加宽度为右下角的横坐标，纵坐标加高度为右下角纵坐标
- 5、调用Image里的open方法打开截图
- 6、使用crop方法按照坐标截图
- 7、Save保存截图

```
1 from selenium import webdriver
2 from PIL import Image
3 url="http://192.168.2.2:8080/bos/login.jsp"
4 driver=webdriver.Firefox()
5 driver.get(url)
6 #截图并保存
7 driver.get_screenshot_as_file("bos.png")
8 #获取元素的位置
9 loc=driver.find_element_by_id("loginform:vCode").location
10 #获取元素的大小
11 kw=driver.find_element_by_id("loginform:vCode").size
12 x=loc['x']
13 y=loc['y']
14 xx=loc['x']+kw['width']
15 yy=loc['y']+kw['height']
16 #打开截图
17 a=Image.open('bos.png')
18 #按照坐标截图
19 im=a.crop(x,y,xx,yy)
20 #保存截图
21 im.save("code.png")
```

## 18、怎么导出html自动化测试报告

- 1、首先引入HTMLTestRunner这个外部文件
- 2、格式化时间为年月日时分秒的格式来定义报告的文件名及定义测试报告的存储路径
- 3、采用open函数，以wb(二进制)的方式来读取这个文件
- 4、采用HTMLTestRunner这个类传入打开的文件，测试报告的标题，测试的详细描述
- 5、执行测试套件，就可以生成测试报告

## 19、UI自动化框架是什么？

python + selenium + unittest + ddt以及POM设计模式

## 20、自动糊跑一个流程要跑多久？

这个取决于不同的项目，项目流程长的运行时间就长，项目流程短的运行时间就短，一般情况下 运行1分钟左右

## 21、自动化主要实现哪些方面的功能？

自动化测试主要用来回归项目主流程

# 第七章（网络相关）

---

## 基础知识

---

### 1、什么是host

（ C:\Windows\System32\drivers\etc ） 访问一个网站的时候会去host查看是否具有配置该域名，如果在host文件配置了该域名的ID，将不会解析该域名，直接访问在host配置的id，如果在host文件没有找到该域名，将会对该域名进行解析然后进行访问（提高常用网站的访问速度）

### 2、禅道（bug管理工具）

如何提交bug、包含产品的选择、指派，bug的优先级，bug的优先级，bug的类型  
提bug给开发-->开发解决-->测试指回去-->开发重新解决-->测试关闭bug-->测试重新打开bug-->开发解决-->测试关闭bug

#### 3、如何添加产品，添加项目，上传项目的相关文件

#### 4、开发如何解决bug

#### 5、如何关闭bug

#### 6、如何把bug重新打回去

#### 7、如何重新打开bug

#### 8、其他平台（mantis、jira、redmine）如何提bug，提bug的步骤

#### 9、如果网页打不开

- 1、没有网（检查网线，检查网络是否正常）
- 2、浏览器有故障
- 3、网址是否正确

- 4、服务器是否拒绝请求
- 5、检查防火墙是否打开
- 6、检查是否配置了host
- 7、浏览器是否授权使用网络
- 8、浏览器版本与网页不兼容
- 9、网络比较差导致请求超时
- 10、服务器不存在请求的资源
- 11、服务器不支持当前的http协议
- 12、请求中数据丢失
- 13、服务器出现拥堵
- 14、浏览器设置了代理
- 15、本地的资源占用过多
- 16、Nginx出现问题
- 17、Dns出现了问题
- 18、网站证书出现问题

## 2、Jira如何提bug

## 3、Redmin如何提bug

## 4、Mantis如何提bug

## 5、版本控制工具如何下拉库里面的东西（从无到有） (svn/git)

## 6、如何更新本地版本库(svn/git)

## 7、如何把本地版本库的东西上传到服务器(svn/git)

### 1.svn

安装svn 进行汉化 设置语言 进行检出（地址、用户名、密码）可以下拉到我们需要的东西 学会更新、提交

### 2.git

安装git、去网上下载时（不能下载gitblit）进入git网址 点击版本库 点击自己想下拉的项目 复制出页面上的ssh开头的网址 新建或者打开自己的项目存放的文件夹 在文件夹里面右键 点击git bash（会出现git的命令框）接下来是重点 下拉本地原本没有的项目（从无到有的过程）ssh://test001@192.168.2.2:29418/第八期.git 回车 输入密码 点击回车后可以得到，自己想要的项目内容 如果服务端有更新要更新本地的文件 进入到项目文件（注意不是放项目的文件）右键点击git bash 输入 git pull 回

车之后可以下拉到更新后的内容 往服务端提交文件 在想要提交的东西复制到自己项目的文件夹以后 先进行一次 git pull操作 git push 然后再输入 git add. git commit -m '提交的原因' git push origin master (分支名称)

## 8、osi的七层模型、tcp/ip五层模型（知道物理层，传输层、应用层有哪些东西）

osi七层	tcp/ip五层	
应用层		
表示层		
会话层	应用层	HTTP FTP TFTP DNS SMTP Telnet
传输层	传输层	tcp udp
网络层	网络层	IP ICMP RIP IGMP
数据链路层	数据链路层	ARP RARP IEEE802.3 PPP CSMA/CD
物理层	物理层	FE自协商 Manchester MLT -3 4A PAMS

HTTP超文本传输协议	Hyper Text Transport Protocol
https超文本加密传输协议	Hyper Text Transfer Protocol
DNS域名解析、域名服务器	Domain Name Server
Telnet远程登录协议	
FTP文件传输协议	File Transfer Protocol
SMTP简单邮件传输协议	Simple Message Transfer Protocol
NFS网络文件系统服务、网络文件系统	Network File System
TCP传输控制协议	Transmission Control Protocol
udp用户数据报协议	User Datagram Protocol
TFTP简单文件传输协议	Trivial File Transfer Protocol
ICMP网间控制报文协议	Internet Control Messages Protocol
IGMP因特网组管理协议	Internet Group Message Protocol
RIP路由选择信息协议	Routing Information Protocol

## 9、Cookie 和Session 的区别

- 1、Cookie 的数据是存放在客户端的浏览器上，Session 的数据放在服务器上
- 2、Cookie 不是很安全，别人可以分析存在本地的Cookie 进行Cookie 欺骗
- 4、一般情况下，单个Cookie 的大小不能超过4K，很多浏览器限制一个web站点最多保存20个Cookie，一般一个浏览器保存最多300个Cookie
- 3、Session 会在一定的时间内存放在服务器上面，当访问增多的时候，会比较占用服务器 的资源
- 5、一般来说，将登陆信息账号密码以及隐私信息等重要信息存放在Session 里面，其他信息如果需要保留，可以存放Cookie 里面。

## 10、token特性

token（令牌）：一般用于APP的验证

最简单的token是由：uid（用户id），time（当前时间戳），sign（签名）

可以把一些不变的参数放在token里，方便查询

**基于token的验证方式，在服务器端不需要存储用户的登录信息：**

首先客户端使用用户名和密码进行请求登录 服务器收到请求之后去验证用户名和密码 验证成功后，服务端会签发一个token，再把这个token传给客户端 客户端收到token后，把他存储起来，比如说可以存放在cookie里面 客户端每次向服务器请求的时候都要带着服务器签发的token 服务器收到请求之后，然后去验证token是否正确，如果正确就返回数据 不正确就提示重新登录，也会验证有效期

## 11、Tcp和udp的区别：

- 1 | tcp是一种可靠的传输协议，连接时需要三次握手四次挥手，传输数据时服务器必须发回确认，tcp有确认机制、重传机制、拥堵机制等等，比较占用系统资源；而udp没有繁琐的传输机制，所以传输很快，但同时它也不稳定不可靠，比如网络不好容易丢包

Tcp是一种可靠的传输协议，是面向连接的，建立连接是需要三次握手断开连接时需要四次握手，所以tcp是比较稳定的，它规定浏览器向服务器发送数据时服务器必须发回确认，但是他也有缺点，因为tcp在与服务器进行交互的过程中，其中包含了确认机制，重传机制，拥堵机制等等相当于udp来说比较耗费时间，而且每个连接都会占用系统资源。而udp协议没有tcp的握手，确认，重传，拥堵等机制，他是一个无状态的传输协议，所以他在传输的过程中会非常快，但是同时它又是不可靠的不稳定的，比如说在和服务器进行交互的过程中如果遇到网络状况不好的情况会造成数据包丢失。



## 12、什么是三次握手什么是四次挥手（用自己的话说出来）

三次握手：

第一次：主机A发送建立连接的数据包到服务器，主机B接受，A要求建立联机

第二次：主机B收到请求后确认联机信息，向A发送建立连接的确认包

第三次：主机A收到之后检查确认包是否正确，若正确，主机A再发送确认包，主机B收到并确认，然后建立连接成功

四次挥手

第一次挥手：客户端A发送一个结束包，用来关闭A到B的数据传输，客户端进入结束等待状态

第二次挥手：服务器B收到这个结束包，并发回一个确认包，服务器进入等待关闭状态

第三次挥手：服务器B在发送一个结束包用来关闭数据传输，服务器进入最后确认状态

第四次挥手：客户端A收到结束包之后向B发送一个确认包且序号+1，服务器B收到之后关闭连接，A等待2MSL之后A也关闭连接。（2MSL意思是2倍的等待时间）

## 19、http请求方式（8种）：

options , head , get , post , put , delete , trace , connect

### GET

向特定的资源发出请求。

### POST

向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的创建或已有资源的修改。

### DELETE

请求服务器删除Request-URL所标识的资源。

## 13、什么是对称性加密什么是非对称加密（区别）知道MD5

对称性加密（AES加密，DES加密）

加密解密都使用相同密钥

## 非对称性加密(RSA加密)

加密和解密都使用不同密钥（私钥可以推测出公钥，公钥无法推测出私钥）

## MD5

不可逆，压缩性

- 1        **BASE64:**
- 2        编码方式（8位字节代码），二进制与字符串相互转换
- 3        **MD5:**
- 4        Message Algorithm（消息摘要算法第五版），散列函数（哈希算法）\_不可逆，压缩性
- 5        **DES:**
- 6        Data Encryption Standard（数据加密标准），对应算法是DEA
- 7        特点：1. 对称加密 2. 同一个SK
- 8        **AES:**
- 9        Advanced Encryption Standard（高级加密标准）
- 10       特点：1. 对称加密 2. 一个SK扩展成多个子SK，轮加密
- 11       **RSA:**
- 12       特点：1. 非对称加密，即：PK与SK不是同一个
- 13                2. PK用于加密，SK用于解密
- 14                3. PK决定SK，但是PK很难算出SK（数学原理：两个大质数相乘，积很难因式分解）
- 15                4. 速度慢，只对少量数据加密

## 为什么要弱网测试

为了测试软件在弱网情况下会不会出现崩溃，加载失败的情况

## 抓包工具

### 抓包方式

F12 , jmeter , charles , fiddler

### 抓包的目的

主要是定位问题，关注的是请求内容和响应内容（响应结果和预期结果是否一致），抓取request和return，抓的是接口。如果抓包结果与预期结果一致，但前端显示结果与预期结果不一致，说明是前端的问题；如果抓包结果与预期结果不一致就说明是后端的问题

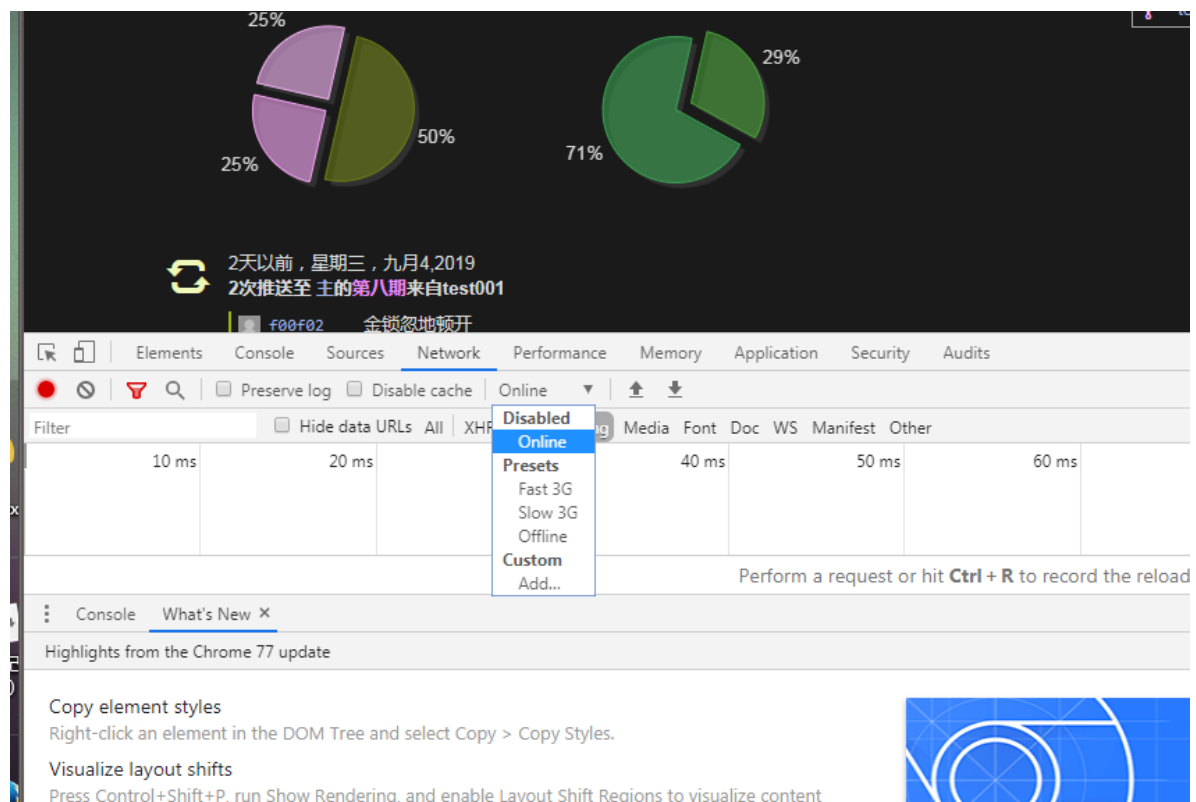
## 14、F12、fiddle、jmeter、Charles分别怎么进行抓包

抓包找到以下内容

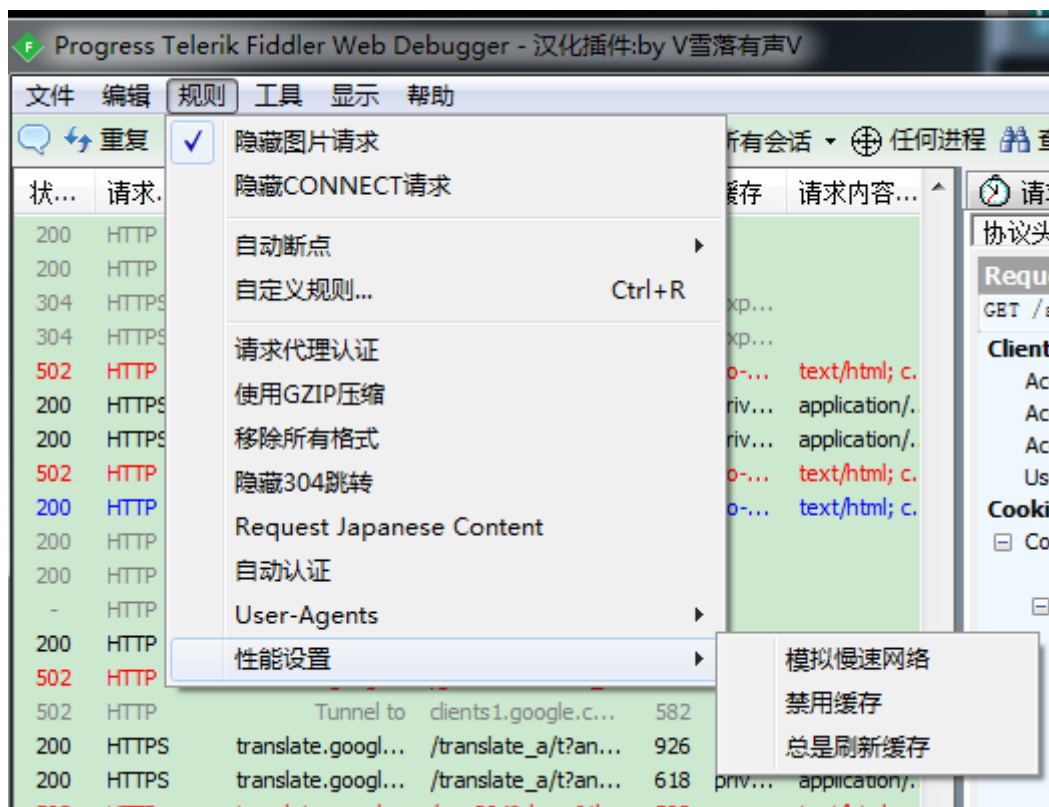
- url
- 请求方式
- 请求头的内容
- 请求参数响应头的内容
- 响应结果

## 15、F12、fiddler、Charles分别怎么进行弱网测试

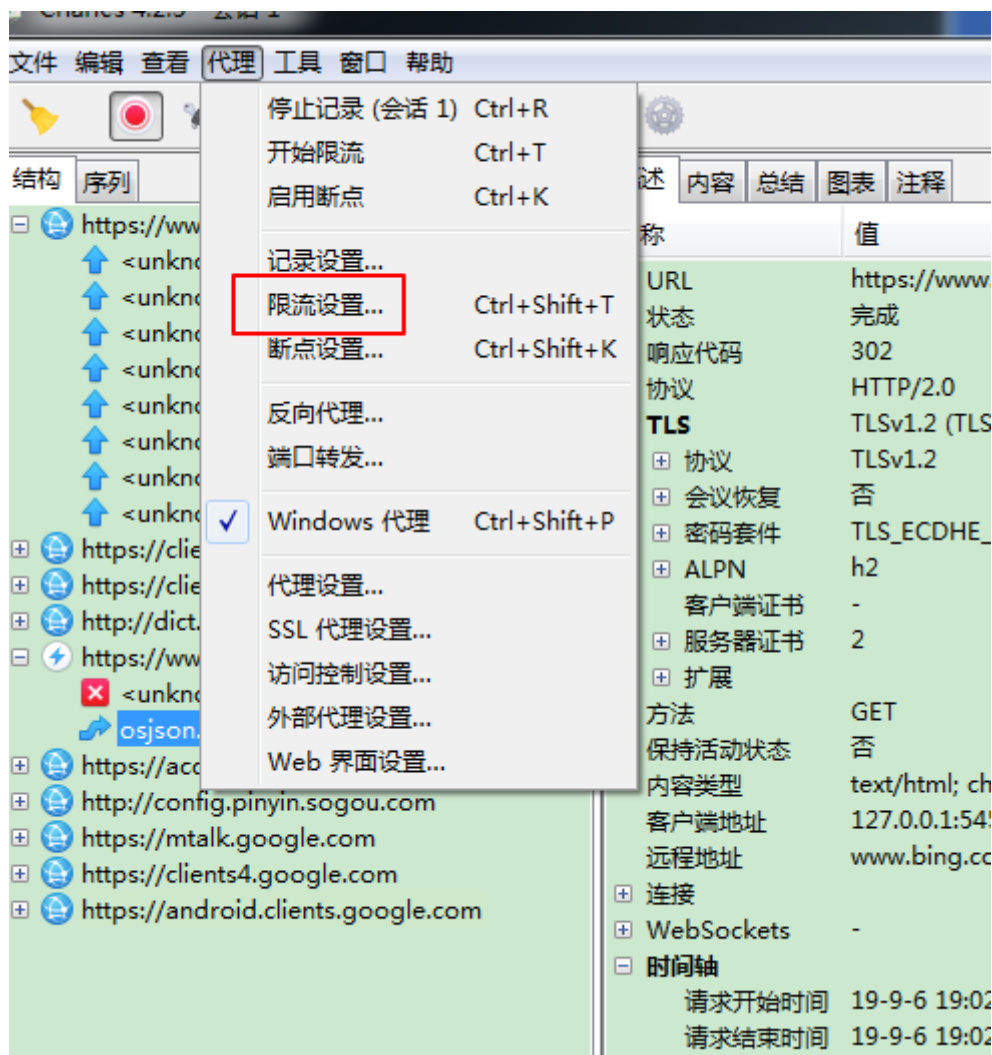
F12



fiddler



charles

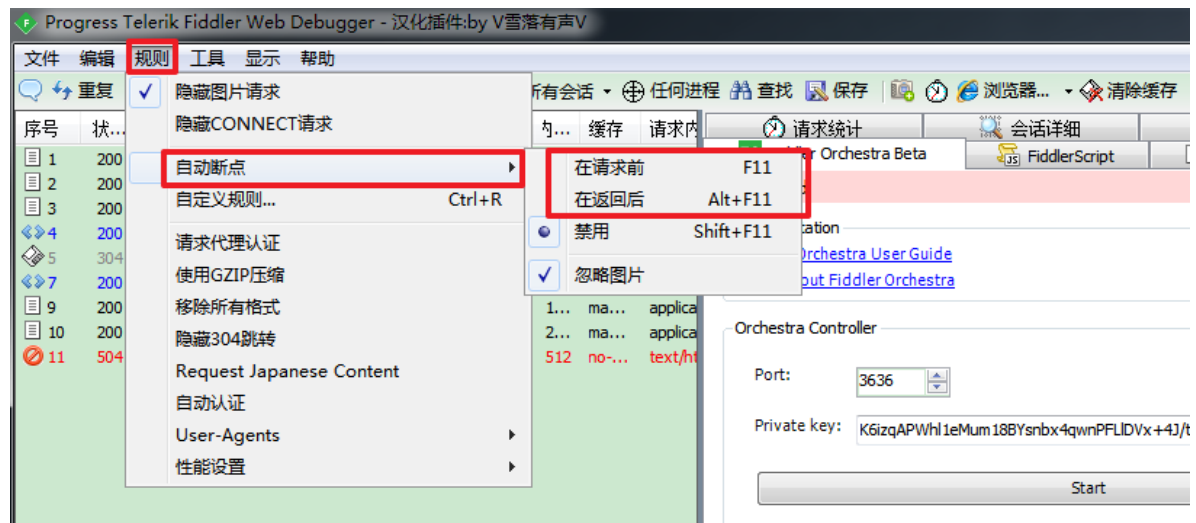


## 16、fiddle、Charles怎么进行打断点

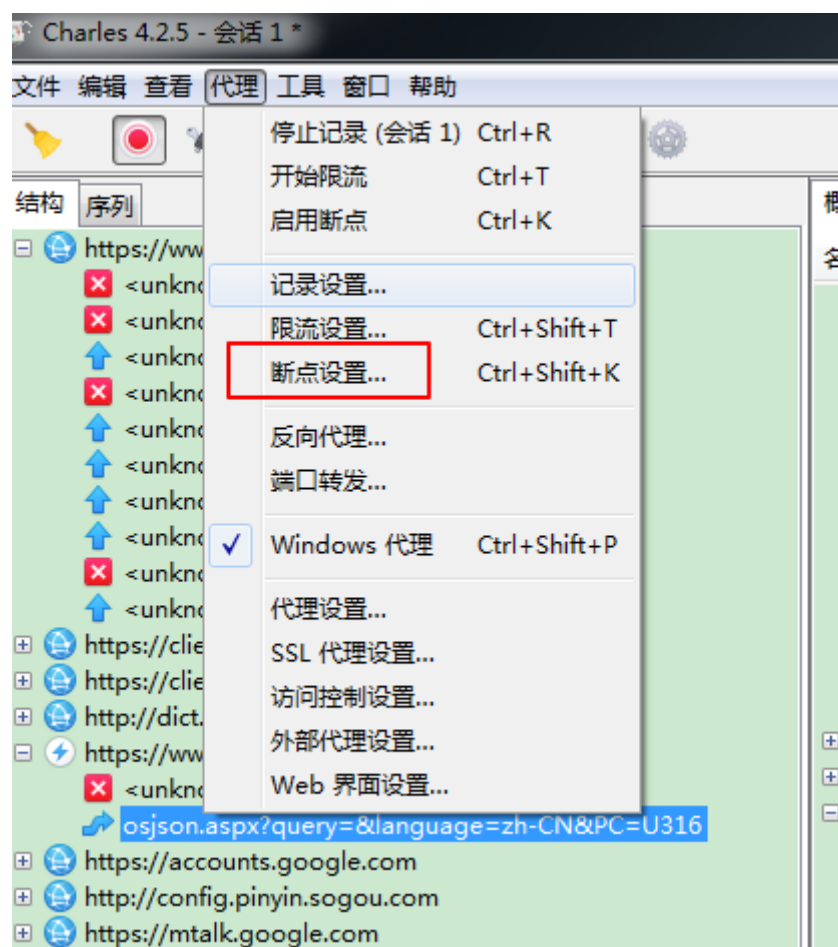
a)打断点

b)改参数 (得到自己想要的结果)

fiddler (提琴手)



charles (花瓶)



## 17、fiddler、Charles怎么进行抓https的包

设置好Windows代理后装相应的证书抓 (端口号: 443)

## 18、fiddler、Charles怎么抓手机的包

手机手动设置代理后再装相应的证书抓（手机连接局域网后设置本机IP代理）

### jmeter

1、打开jmeter

- 在http代理服务器上——配置端口8888 ——配置本机ip
- 目标控制器勾选测试计划>线程组 启动既可

2、在测试计划下面添加线程组

3、在线程组（或者工作台）下面添加http代理服务器

4、https domains 输入127.0.0.1（本地服务器）（domains域名）

5、目标控制器选择是计划线程组

6、点启动

7、设置系统代理

方法一：打开谷歌，打开设置，高级设置---系统---打开代理设置---连接---局域网设置---勾选代理服务器---输入地址---输入端口号---确认

方法二：打开ie 打开设置，internet选项---连接---局域网设置---勾选代理服务器---输入地址---输入端口号---确认

8、打开jmeter 打开网站 就可以查看了

### charles

打开charles 点击代理，勾选windows代理

再打开代理设置把端口设为8888，然后打开网页点击设置选项，点击连接中的区域网设置，然后点开代理服务器把端口，也设为8888，输入自己的ip点击确定，就可以进行抓包了

charles 证书安装

点开帮助点击ssl代理，安装charles根证书，存储到受信任的根证书颁发机构，然后在打开 网页执行，再看抓取的信息，会提示此主机未启用ssl代理

### 如果不能抓HTTPS的包

- 1.打开浏览器选择internet选项，选择连接，局域网设置，去设置
- 2.打开charles,选择代理，代理设置，设置成为8888
- 3.打开代理，windows 代理即可。

- 4.在帮助-ssl代理-安装charles根证书，安装在受信任的根证书颁发机构。
- 5.点击代理，ssl代理设置。启用代理，用:表示。即可

### 限速：

- 1.打开charles
- 2.打开限速(小乌龟形状)可以在代理里面设置限速设置
- 3.设置系统代理(在打开charles的抓包后系统代理自动打开)
- 4.浏览器在打开网上

### 打断点：

- 1.打开charles
- 2.打开断点(在代理里面可以设置断点)
- 3.访问
- 4.即可对接口进行修改(如果断点设置为请求打断点可对请求进行修改，然后发送到服务器，如果断点设置为响应即可对响应结果进行修改，然后点击执行后把修改后的内容返给浏览器)
- 5.在结构栏里面可以右键点击设置打断点的对象

### 过滤的三种方法：

- 1、结构下右键网址点重点
- 2、序列下在过滤框输入想过滤的内容
- 3、在结构下面的过滤框输入想过滤的内容

### 对手机进行抓包

1. 先打开charles
2. 必须在同一局域网ping ip通
3. 手机代理设置为要抓包电脑的ip端口号
4. 自带浏览器打开网址 chls.pro/ssl
5. 加密与凭据安装证书（苹果手机：设置-通用-关于本机-证书信任设置-找到 charles proxy custom root certificate ）
6. 之后就可以抓手机包了

## fiddler

if (m\_SimulateModem)设置搜索延时

## 4、断点

### (1)全局断点

全局请求的断点 全局断点就是中断fiddler捕获的所有请求，先设置下，点击rules-> automatic breakpoint ->before requests 全局响应的断点 全局断点就是中断fiddler捕获的所有请求，先设置下，点击rules-> automatic breakpoint ->after responses

## (2)非全局断点

先取消全局断点 打全局断点的话，是无法正常上网的，需要清除断点：rules-> automatic breakpoint ->disabled 请求断点：bpu（打断点加域名，不打不加域名直击bpu回车） 请求加响应断点：bpafter（打断点加域名，不打不加域名直击bpafter回车）

## 1、抓http的包

- 1 打开fiddler后--tools--winINET Options--设置网络代理
- 2 Tools---Options--connections--设置抓包的配置

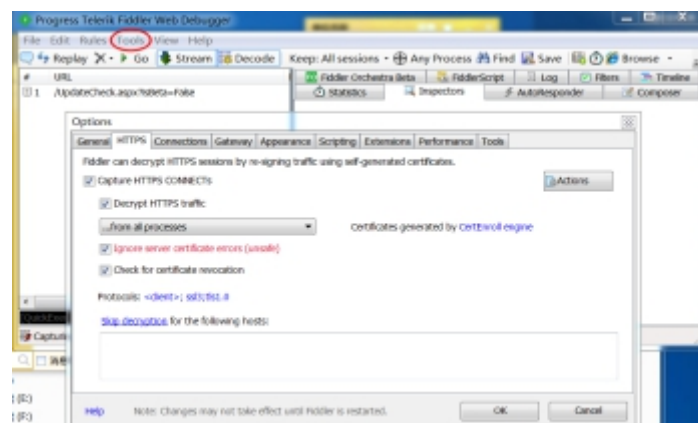
## 2、抓https的包

(1) 对于https的协议是带有安全的功能，所有有一些的https的协议是无法抓到的，所以需要通过设置fiddler中来对，来使用fiddler的方式的来抓取到https的请求包，那么如何使用fiddler中抓取到请求包。

(2) 进行打开电脑中已经安装好的fiddler的软件，进行点击菜单中“tools”的选项菜单。

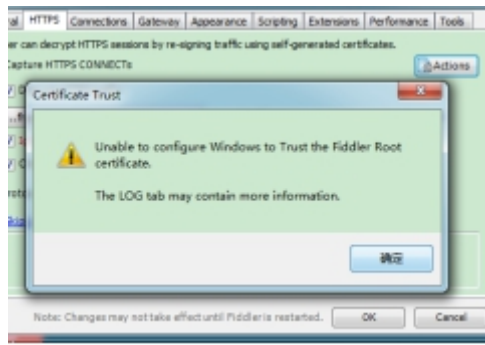
(3) 这样就会弹出了下拉菜单中进行选择为“fiddler option”的选项的菜单

(4) 这样就会进入到了fiddler options的选项窗口中的，进行点击菜单中的“HTTPS”的选项。



如果在安装证书的时候出现错误的话





### 3、弱网测试

- (1) 在fiddlerscript里面设置网速if (m\_SimulateModem)设置网速
- (2) 在rules里面打开弱网测试performance里面的simulate modem打开弱网测试

### 4、断点

#### (1) 全局断点

##### 全局请求的断点

全局断点就是中断fiddler捕获的所有请求，先设置下，点击rules-> automatic breakpoint ->before requests

##### 全局响应的断点

全局断点就是中断fiddler捕获的所有请求，先设置下，点击rules-> automatic breakpoint ->after responses

#### (2) 非全局断点

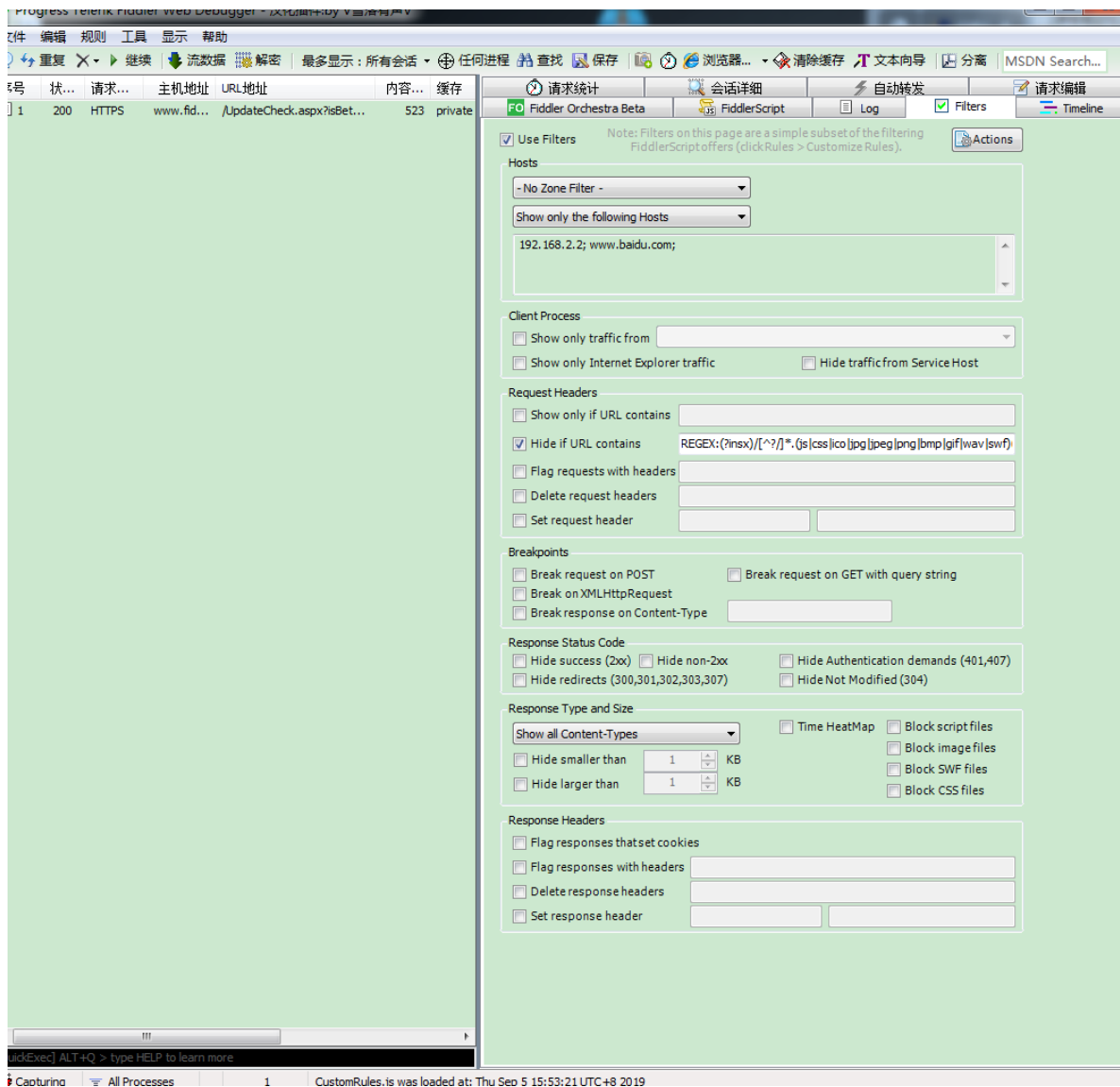
- 1 先取消全局断点
- 2 打全局断点的话，是无法正常上网的，需要清除断点：rules-> automatic breakpoint ->disabled
- 3 请求断点：bpu（打断点加域名，不打不加域名直击bpu回车）
- 4 响应断点：bpafter（打断点加域名，不打不加域名直击bpafter回车）
- 5 两个同时：先bpu https://www.baidu.com再bpafter https://www.baidu.com

### 5、手机抓包

### 6、过滤

- 1 192.168.2.2;www.baidu.com;

- 1 REGEX:(?insx)/[^\?/]\*.  
(js|css|ico|jpg|jpeg|png|bmp|gif|wav|swf)(?\.\*)?&



## 20、http状态码代表的意思

# APP测试与web测试

## 21、APP测试的测试点

- 1 功能测试、接口测试、性能测试、UI测试、安装和卸载测试、升级和更新测试、权限测试、安全测试、网络测试、中断测试、手势操作测试、消息推送测试、兼容性测试、渠道包测试、软件占用系统资源的测试

### 1、功能测试

主要关注软件功能是否正常，是否符合设计需求

### 2、接口测试

主要测试各模块之间的接口是否正常，数据传输是否正确

### 3、性能测试

主要关注app的压力和负载测试

#### **4、ui测试**

用户界面（比如：菜单，列表，对话框）等页面布局，是否满足客户的需求，界面文字以及图片显示是否正确，跟产品模型是否一致

#### **5、安装和卸载测试**

不同版本是否可以重复安装

旧版本是否可以安装

安装路径是否正确

卸载的时候是否会提示保留用户信息

卸载之后是否会有数据残留

卸载之后再次安装是否保留信息

#### **6、升级更新测试**

更新之后数据是否正常

是否会提示更新

是否可以手动、自动更新

优化更新之后是否有数据残留

跨版本更新

是否可以版本回退

更新遇到突发情况

#### **7、权限测试**

软件获取通讯录、相机、照片、地址、麦克风、网络等权限是否可以正常开启和关闭

获取第三方登录，比如：软件是否正常微信、QQ、支付宝等第三方登录

不同用户权限测试，比如普通用户和vip用户

#### **8、中断测试**

主要关注APP在运行过程中突然来电话来信息

以及文件下载和听音乐的操作时中断情况

#### **9、手势操作测试**

下拉刷新、上划加载，左右前进返回测试，放大缩小等

分屏测试、杀死后台测试 前后台测试切换

## **10、推送消息测试**

站内信是否可以正常推送

是否可以给指定的用户推送消息

特别推送信息是否可以正常推送

不接受消息是否不再推送

免打扰是否不再推送，正常时间是否可以补发消息

## **11、安全测试**

主要关注用户的隐私，授权级别，敏感数据等

## **12、网络测试**

主要关注2G/3G/4G以及WiFi/不同网络格式比如联通电信移动下软件的运行情况，以及在网络不好的情况下软件是否会出现崩溃（crash）

### **查看崩溃日志**

Android:/storage/emulated/0/UZMap/log/crash

ios:/documents/APIcloud/savaFile/exceptioninfo

## **13、兼容性测试**

首先考虑不同平台：Android和iOS

考虑不同系统版本： 安卓：4.0/5.0/8.0 iOS：10/11/12

考虑不同手机型号 安卓：小米、华为、三星、ov等

ios:6/6s/6p,7/7s/7p,8/8p,iphone x等

不同手机屏幕尺寸：720p、1080p、2k

我们主要关注APP在不同兼容性下各种功能及界面显示是否正常，符合设计需求等

### **兼容性测试的时间点**

在APP的各种功能都正常之后在回归测试之前我们会进行APP的兼容性测试

## **14、渠道包测试**

开发主要是根据APP发布的不同渠道进行打包，比如说安卓市场、应用宝、豌豆荚等，不同渠道有不同的渠道包，我们需要注意的是验证发布的渠道是否与对应的标识一致，以及发布之后是否可以正常的安装和更新。一般情况下开发是把不同的渠道标记在代码里，我们核对一下对应渠道与标识是否匹配就可以了。

## 15、软件占用系统资源的测试

主要关注功能测试、性能测试、接口测试、UI测试、安装测试、升级更新

## Web 测试的关注点

除了最基本的功能、性能、接口、UI、网络等测试，我们一样还要考虑兼容性。主要关注不同浏览器比如IE(8/10)，Firefox(6.0/7.0)，Chrome(65/66/77)等主流浏览器的兼容性，考虑不同浏览器的版本是否能够兼容，以及在不同浏览器下面图片，页面布局，页面样式是否能够正常显示、功能按钮是否能够正常使用。

### H5和原生APP的区别：

手机端的web页面，H5又称为webapp，原生APP又称为 native app 首先原生app根据不同的平台需要分别开发，相当于C/S架构。而webapp相当于B/S架构，是基于浏览器的不需要单独开发，比如手机浏览器访问网站，手机浏览器访问页面。

## 23、Web测试和APP测试有什么区别？

首先web测试跟app测试在测试流程上没有什么区别，我个人觉得是由于载体不一样，所以在系统测试时会有一些细节不一样，Web项目一般都是B/S架构的，是基于浏览器的，而app是C/S架构，必须有客户端。

### 从系统架构上面来说，

web测试只是更新了服务器端，客户端就会同步更新 而APP端如果修改了服务器端的资源或者系统架构，客户端所使用的所有核心版本都需要回归测试一下

### 然后从兼容性方面

首先web是基于浏览器的，一般测试的构成中以浏览器的兼容性为主，我们一般考虑是不同内核的浏览器，比如说IE，谷歌，火狐等浏览器，而APP的兼容性必须依赖手机或pad，我们在测试的过程中，不仅要考虑不同的平台，主要就是安卓和iOS，而且要考虑主流的机型，其次是不同的手机型号，系统版本，屏幕尺寸和分辨率等，相比较web测试APP测试又多了一些专项测试比如权限测试，安装卸载，升级更新，中断 网络手势 和消息推送以及占用系统资源的测试 总的来说，测试的产品千变万化但测试的思维是不变的

## 22、定位问题的四种方法

动态日志、抓包、状态码、数据库

## 28、弱网测试的方法

### 第一种：使用360WiFi，WiFi共享精灵等工具去控制

对应设备的网速，以测试APP在弱网情况下的表现

### 第二种：使用chrome浏览器的控制台控制网络速度

在network里面的offline和online我们可以控制加载的速度 offline（离线）  
online（在线） slow3G（慢3G） fast3G（快3G）

### 第三种：使用fiddler进行弱网控制

首先我们要配置好代理 在fiddlerscript里面修改数据上传下载的延迟速度 然后在rules（规则）>performance（性能设置）里面有一个simulate modem speeds（模拟慢速网络）打开这个弱网测试开关 然后就可以模拟上传下载的速度进行弱网测试

## 24、App测试与H5测试的区别

## 25、小程序测试与H5测试的区别

## 26、如果网页打不来如何排查问题

## 27、Xml、html、JavaScript、json

## 29、http与https的区别

https协议是由ssl+http协议构成的可进行加密传输、身份认证的网络协议，要比http协议安全

## 1、APP出现anr是什么原因导致的

1. 耗时的网络访问
2. 大量的数据读写
3. 服务器繁忙导致连接超时无响应
4. 其他线程持有锁，导致主线程等待超时
5. 其他线程终止或崩溃导致主线程一直等待

## 2、APP出现crash有哪些原因？

1. 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APPcrash
2. 是内存泄漏，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有可能会导致APPcrash
3. 程序逻辑错误：数组越界、堆栈溢出、并发操作、逻辑错误

4. 设备兼容：由于设备多样性，APP在不同设备上可能会有不同表现
5. 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APPcrash。或者是不同网络的切换也会影响APP的稳定性

### 3、APP对于不稳定偶然出现anr和crash的时候你是怎么处理的

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

### 你平常会看日志吗，一般会出现哪些异常？

- 1 | NullPointerException -空指针异常
- 2 | ClassCastException -类型强制转换异常
- 3 | ArithmeticException -算术运算异常
- 4 | NumberFormatException -数字格式异常
- 5 | IllegalArgumentException -传递非法参数异常
- 6 | IndexOutOfBoundsException -下标越界异常
- 7 | UnsupportedOperationException -不支持的操作异常

#### error：

是系统中的错误，程序员是不能改变的和处理的，是在程序编译时出现的错误，只能通过修改程序才能修正，error表示水池漫出来或者水池漏了

#### warn：

提示性错误

#### exception：

表示程序可以处理的异常，可以捕获且可能恢复 exception表示水池有杂志水不好了

## 小程序测试

一个用户只有一个user\_id，但一个用户可以有多个open\_id, 两个小程序之间相互跳转必须绑定同一个公众号，否则无法跳转

## 小程序版本

体验版、开发版、正式版

小程序的测试我们除了要考虑最基本的功能测试，接口测试和页面UI样式测试 业务流测试和数据流测试外有自己的一些特有的测试 首先小程序有自己的一些特点：

1) 他分为三种版本类型：开发版，体验版，正式版 开发版和体验版无需审核，通过扫小程序二维码才能访问 开发版和体验版的区别在于，在开发版小程序二维码有

效期比较短。正式版需要通过微信审核流程。

## 1.权限测试

### 1) 授权测试

用户首次访问小程序，主动获取微信授权，通过获取openid，生成转转uid，并存储昵称、头像等信息。后续用户若杀掉小程序重新进入，无需重新授权。

### 2) 未授权微信登录小程序

未授权时会首先弹出权限提醒，授权之后才可访问小程序

### 3) 已授权微信登录小程序

授权微信访问小程序，意味着自己的微信账号可被小程序管理方所获取，自动以微信的身份行使业务操作权限，比如咨询、支付、数据查询等

### 4) 同一微信号在不同手机端登录授权查看数据权限

### 5) 重新授权

### 6) 取消权限之后再次重新授权

## 4.兼容性测试

1) 手机操作系统 常规的手机端OS为：Android ( 8.x/7.x/6... ) 、IOS ( 11.x/10.x/9.x... )

2) 微信版本测试

3) 小程序码的兼容性测试，因为目前小程序不支持直接分享朋友圈，只能分享微信好友。

4) 屏幕兼容性测试 比如4.0/4.7/5.0/5.7等等屏幕尺寸

## 5网络测试

1) 在wifi和2G/3G/4G网络下是否可以正常访问

2) 网络切换的测试比如从2G/3G/4G切换为wifi

3) 无网络测试以及从有网到无网测试

## 6.更新测试

更新有2种机制

1) 未启动时更新（后台静默更新到最新版本）



2) 启动时更新 ( 每次冷启动都会检查是否有最新版本，如果有就异步下载更新包，并先使用老的包进行启动，新版本会在下次冷启动时应用 )

## 7. 页面样式测试

查看页面样式/ui交互/导航栏/轮播图/背景图/字体等测试

同一公司下多个公众号之间需要用户帐号互通，微信开放平台提供了UnionID机制在不同的公众账号下OpenID是不一样的，而他们的UnionID却是一样的 UnionId和uid是对应关系；OpenId和sourceld是对应关系。

## H5与APP的比较

H5和app其实是产品的不同展现形式各有优缺点 H5比较轻便，易于传播，不用下载一般是通过链接的形式在浏览器中访问，他在各种浏览器、甚至app内传播，而APP只能用户自己在应用官网上下载传播 在手机上安装不能在网页中打开。现在很多app都是原生app结合H5混合开发的模式 这样数据能够更灵活节省开发资源。

H5测试一般除了最基本的功能，接口，性能和UI测试之外也有自己的一些特有的测试

- 1 登录模块
- 2 翻页、返回和刷新
- 3 屏幕旋转
- 4 网络测试
- 5 点击滑动测试
- 6 兼容性
- 7 加载速度
- 8 链接分享

### 1. 针对登陆模块

- 1) 登陆app那么进入H5也应该是登陆状态
- 2) 若app未登陆点击对应的按钮应该触发单点登陆提示登陆，以及取消之后再次点击是否提示登陆
- 3) 页面session或cookie过期是否提示重新登陆

### 2. 翻页/返回和刷新

- 1) 在一页和多页的情况下数据展示是否正常能否翻页跳页
- 2) 翻页之后页面的焦点是否定位正确

- 3) 点击页面内的返回按钮返回逻辑是否符合业务
- 4) 点击物理返回按钮是否符合业务
- 5) 单次刷新能否正常。多次连续下滑刷新是否正常

### **3.屏幕旋转**

### **4.网络切换弱网测试**

在2G/3G/4G和WiFi情况下 是否能够正常访问 以及2G/3G/4G和WiFi的切换情况下是否能够正常

### **5.滑动定位是否准确**

手指滑动是否流畅，手指点击时焦点是否定位正确

### **6.H5链接分享测试**

- 1) 链接分享标题是否正确
- 2) 分享之后数据/样式是否正确合理
- 3) 二次分享以及分享之后是否可以用户操作
- 4) 以及点赞评论浏览量测试

### **7.兼容性测试**

- 1) 不同操作系统兼容性比如安卓4.0/5.0/6.0和ios 8-10-12
- 2) 不同手机型号的兼容性比如小米/三星/华为/魅族 和苹果6-7-8-x
- 3) 不同分辨率比如720/1080/2k屏幕
- 4) 手机浏览器包括自带的浏览器/chrome浏览器/UC浏览器等

### **8.加载速度**

- 1) 不同网络下的加载速度
- 2) 首次打开的加载速度，没有缓存和清理缓存之后的加载速度
- 3) 加载资源的处理比如图片压缩或者放到CDN（内容分发缓存服务器）

打断点的时候（fiddler、charles）

改参数（得到自己想要的结果）

## **第八章（接口测试）**

---

# 简介

---

## 什么是接口测试：

接口测试是测试系统组件间接口的一种测试。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。测试的重点是要检查数据的交换，传递和控制管理过程，以及系统间的相互逻辑依赖关系等。

## 接口测试的原理：

模拟客户端向服务器发送请求报文，服务器接收请求报文后对相应的报文做处理并向客户端返回应答，客户端接收应答的过程

## 接口文档都包含哪些内容：

包含请求地址，调用方式，请求参数，请求参数详情，返回参数，返回值详情。

## 什么时候开始接口测试？

一般后端开发成功前端还没开发完成时候我们可以模拟前端发送请求去测试后端。或者前后端都开发完成了我们可以在功能测试之前先进性接口测试。

## 为什么要做接口测试？

- 1、现在很多系统前后端架构是分离的，从安全层面来说，只依赖前端进行限制已经完全不能满足系统的安全要求因为绕过前端实在是太容易了，需要后端同样进行控制，在这种情况下就需要从接口层面进行验证。
- 2、因为接口中大部分传输的内容是数据，所以说我们要测试数据的准确性，以及数据流的走向，业务走向是否正确。
- 3、接口测试能够覆盖到前端功能测试不能覆盖到的一些点，而且在功能测试层面，接口测试效率更快，测试覆盖相对更全面。能够更高的提高测试覆盖率，增强软件的可靠性。

## 接口测试的流程：

- 1、首先了解需求，了解接口的业务场景，以及尽可能多的细节。
- 2、指定测试方案，除了功能以外，还可能需要考虑性能和安全性要求，根据这些需要设计测试用例。
- 3、然后进行测试用例的review评审：发给相关的开发，开发组长，测试，测试组长，不断根据大家的意见改进，直到大家都同意通过。

- 4、在测试方案review的工程中开始准备测试数据，测试工具，或者是测试用例
- 5、执行接口测试，提交bug验证修复的bug并发送测试报告，确定是否可以上线。

## 接口测试用例是怎么设计的：

一般情况下接口测试用例设计的方法其实跟功能测试用例设计方法是大致类似的，因为接口测试也属于功能测试的范畴，其一样需要满足需求文档里面所规定的功能要求，但是其也有特有的部分，比如说我们设计接口测试用例的时候我们除了需要考虑功能测试最基本的边界值，等价类，因果图，错误推测，场景设计等方法之外还需要考虑

- 1、首先是冒烟测试，说白了就是传递正确的参数，是否返回正常的结果。
- 2、参数组合，因为参数有必传和非必传，参数的类型和长度，以及传递时可能业务上的一些限制，所以在设计用例时，就要排列组合这些情况，保证所有情况都能覆盖到。
- 3、绕过验证，比如提交订单时，在传递商品价格参数时，修改商品价格，就要看后端有没有验证了。或者我支付时，抓个包将订单金额一改，如果能以我改后的金额支付，那这个接口就有问题了。
- 4、权限验证，就是某个功能只有有特殊权限的用户才能操作，那我传递一个普通的用户，看普通用户能否操作
- 5、参数是否加密，这个关系到一些账户的安全，比如我们在登陆一些网站时，它要将我们的登录信息进行加密。
- 6、最重要的我们还要考虑业务逻辑校验，以及请求方式、格式、请求地址、参数、响应结果、响应值的格式，还要考虑数据库的落地情况，比如一些新增更新的接口，查询接口要考虑数据返回的是否完整，与数据库数据进行对比。

## 常见接口的协议类型：

http/https/webservice/dubbo

## 常见接口的请求方式：

Get/post/put

## 接口测试常见的测试工具：

jmeter/soapUI/postman/loadrunner等等

## 有哪些常见的接口：

- 1、通过飞猪订不同航空公司的飞机票，飞机票的信息一般就是通过各大航空公司的接口获取到的。
- 2、淘宝物流信息，一般淘宝物流信息就是通过各个物流公司的接口获取到的。
- 3、第三方微博客户端，个人用户的微博等信息都是通过微博的接口拿到的。

## GET请求与POST请求的区别

1. get和post本质上都是tcp连接，并无差别，但是由于HTTP规定和服务器的限制，导致它们在应用过程中有一些不同；
2. get参数通过url传递，post则放在请求体里边；
3. get请求在URL中是有长度限制的，而post理论上是无限制的，其限制作用的是服务器的处理能力；
4. get相对post来说，get不安全，其参数是暴露在url里，所以不能用来传递敏感信息，如果post传递敏感信息在不加密的情况下也是不安全的；
5. 对于get请求而言浏览器也会把请求头和数据一起发送出去，而post请求是浏览器先发送请求头服务器响应100，然后浏览器再发送请求内容。

## 操作

---

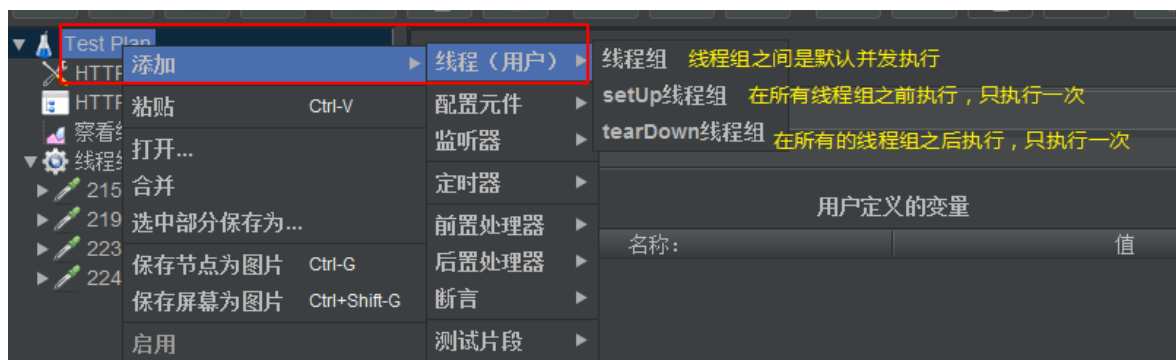
### 1、jmeter脚本录制

- 1、打开jmeter
- 2、添加http代理服务器
- 3、添加线程组
- 4、设置抓包文件储存路径
- 5、打开代理服务器
- 6、打开浏览器代理
- 7、启动代理服务器
- 8、打开要录制的网页
- 9、进行录制脚本

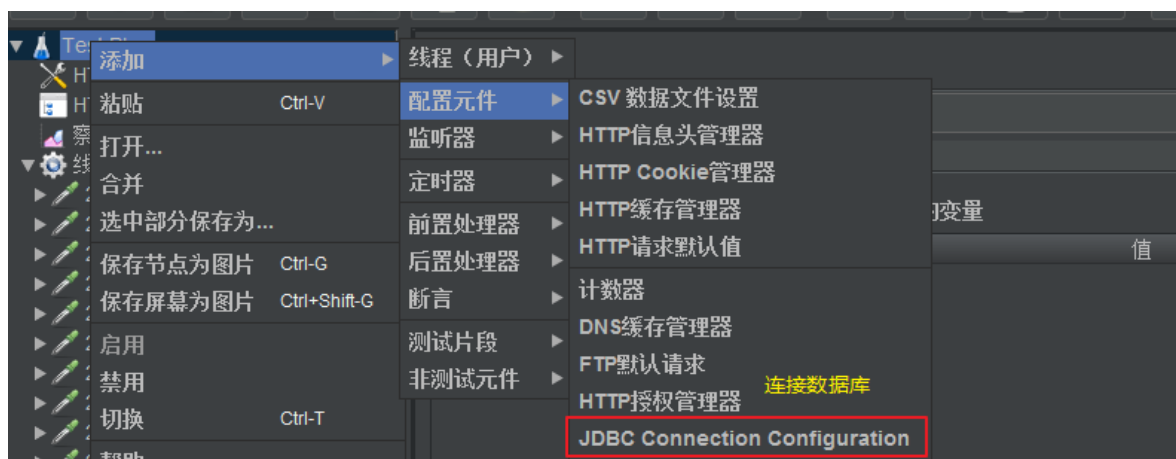
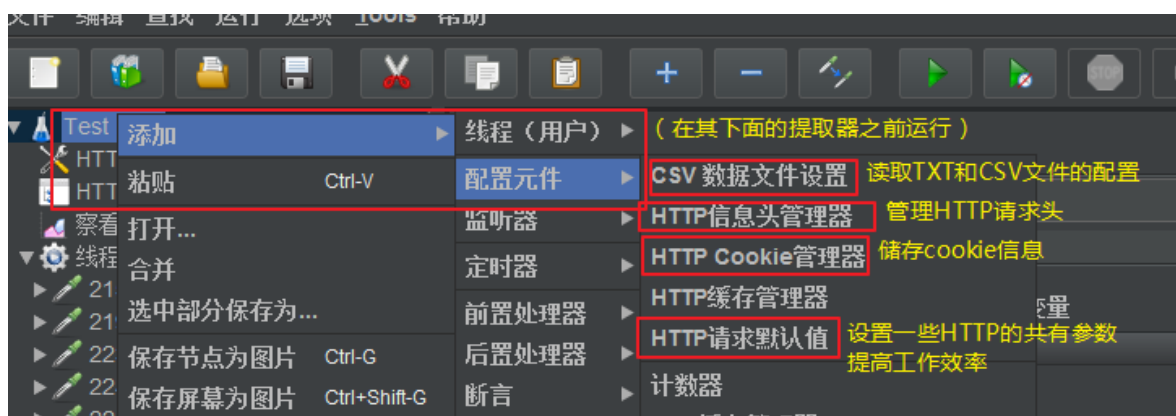
### 2、常用功能

#### 1、测试计划

##### 1、线程



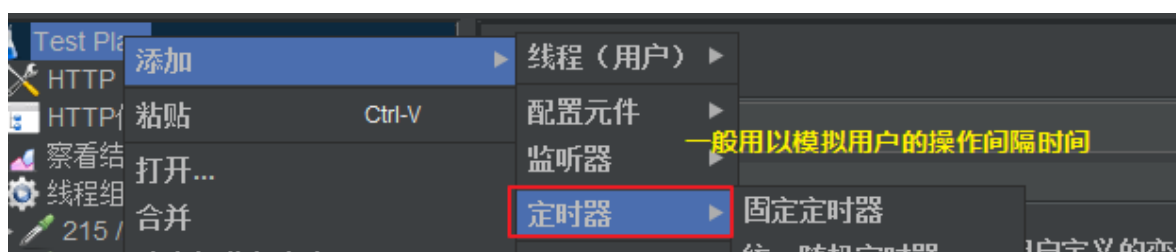
## 2、配置元件



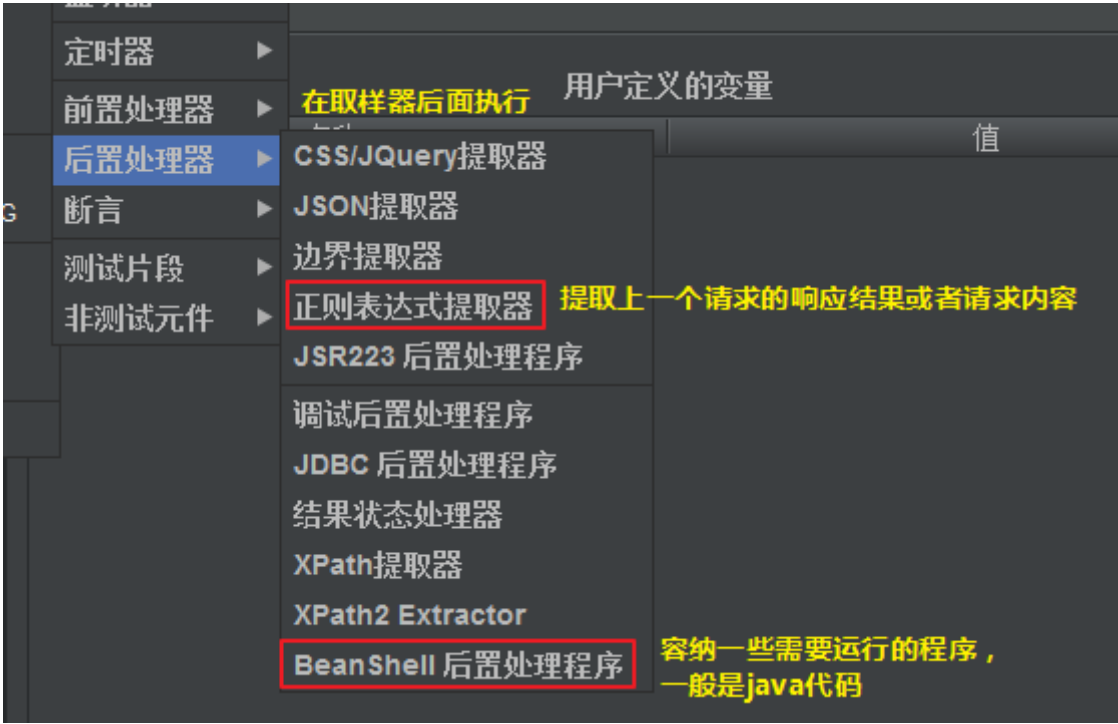
## 3、监听器（监控取样器）



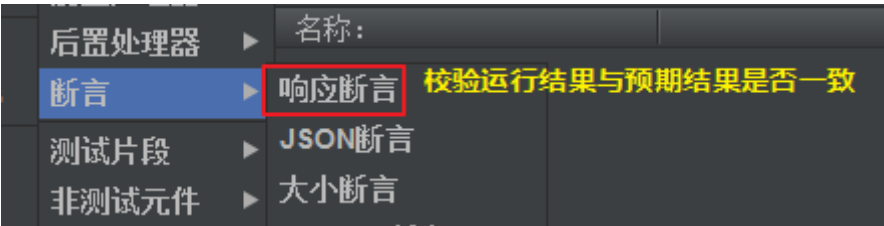
## 4、定时器



5、后置处理器

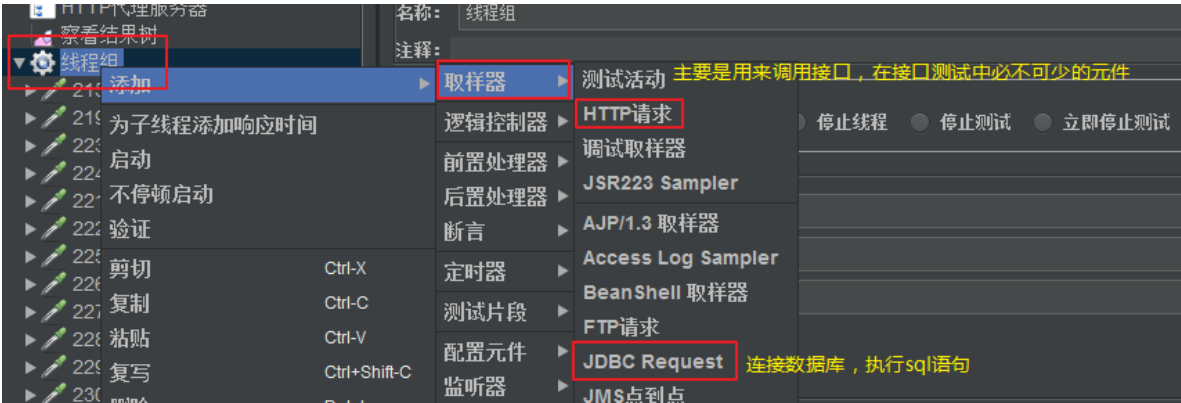


6、断言



2、线程组

7、取样器

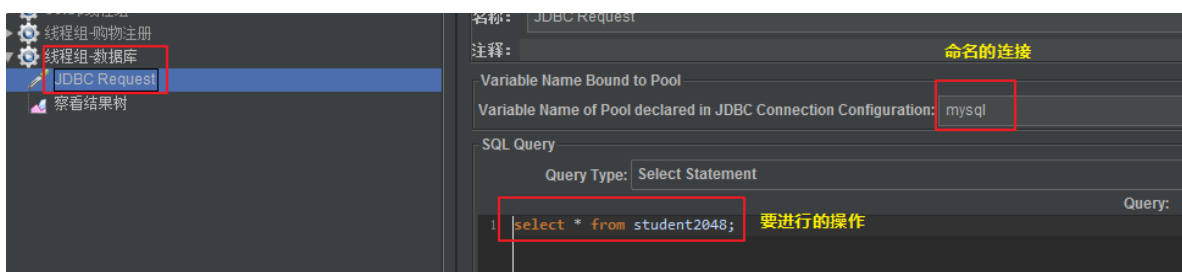
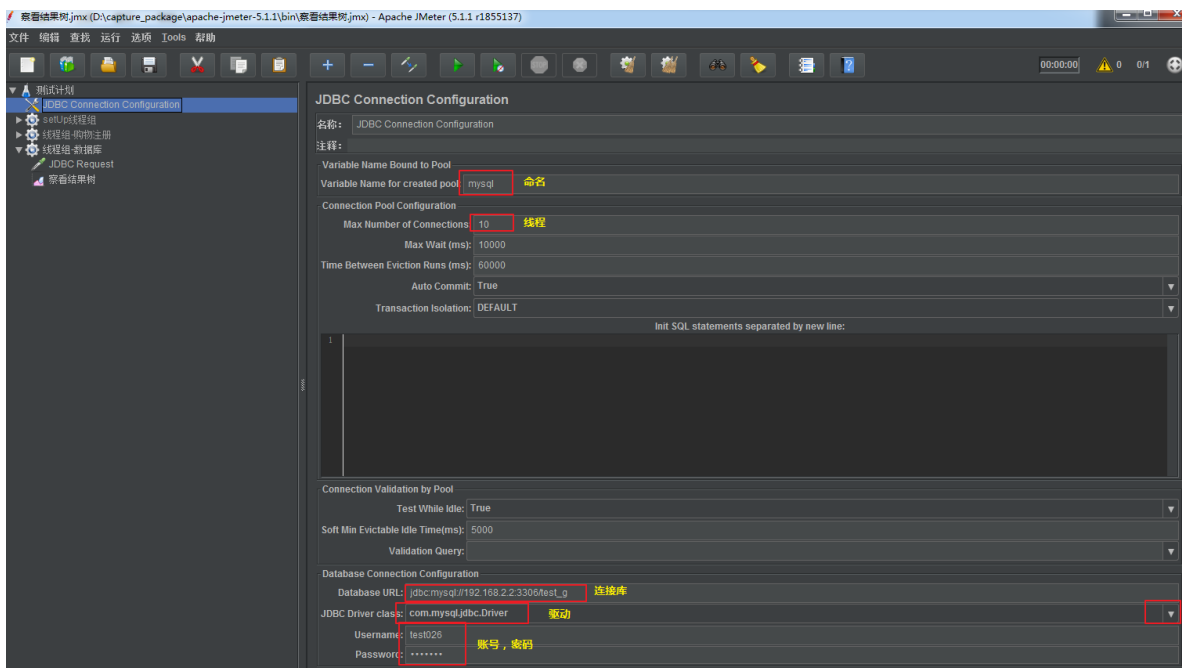
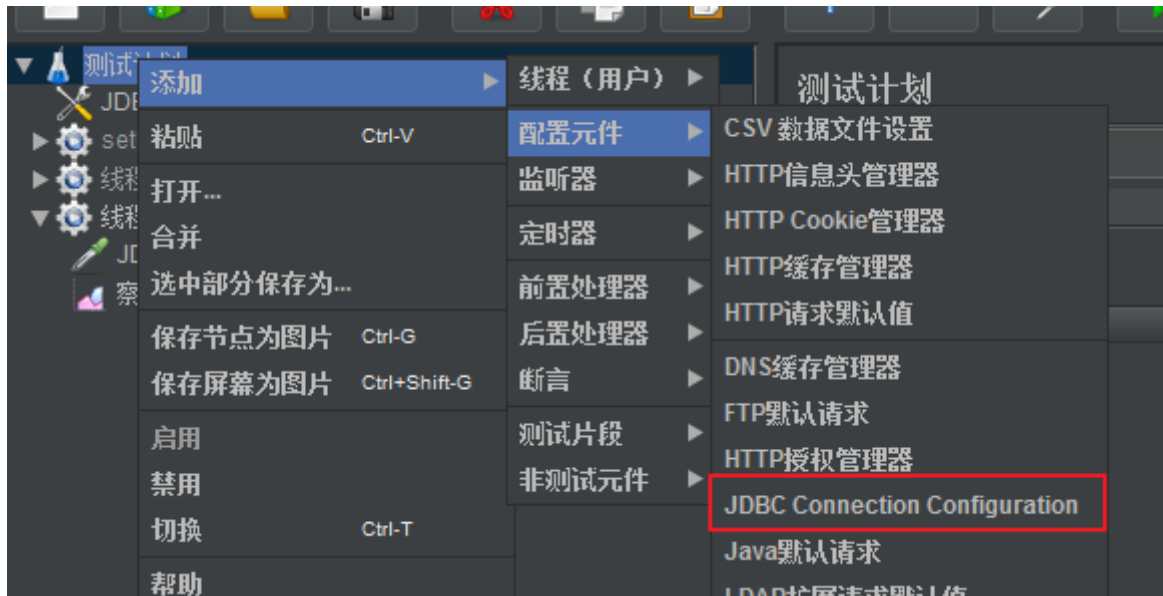


8、逻辑控制器



### 3、使用正则表达式

### 4、jmeter连接数据库



### 5、跨线程组传递数据（ BeanShell ）

1、前提条件：有两个线程组用以传递数据，两个线程组里面分别有一个彼此相关联的接口，怎么从上一个线程组传递数据给县一个线程组

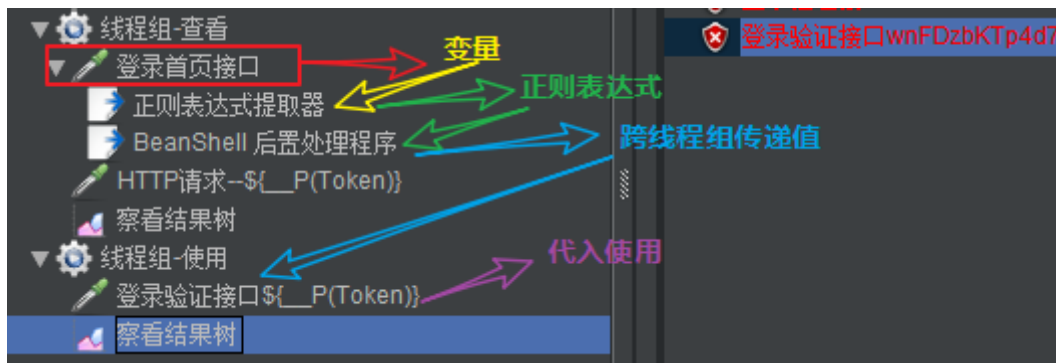
2、在上一个线程组的请求下面添加正则表达式



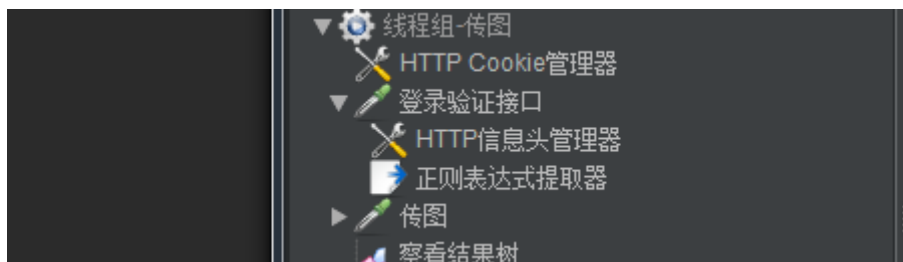
- 3、然后再添加BeanShell 后置处理程序
- 4、填写正则表达式的内容取到想要取到的参数用以传递到BeanShell 后置处理程序里面
- 5、填写BeanShell 后置处理程序相关的参数
- 6、在下一个线程组里的请求里面引用参数

## 6、Jmeter学习内容回顾

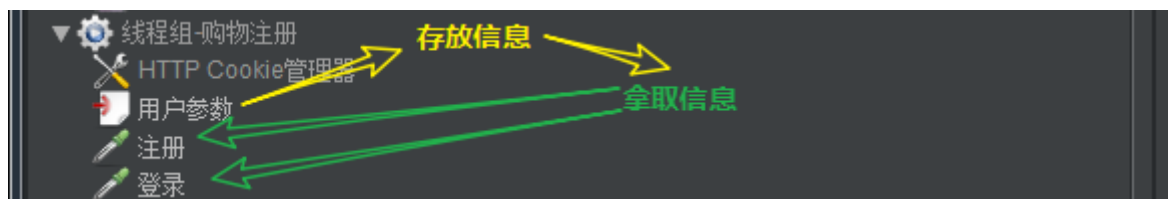
- 1、两个相关联的接口传递数据



- 2、登录接口添加cookie管理器



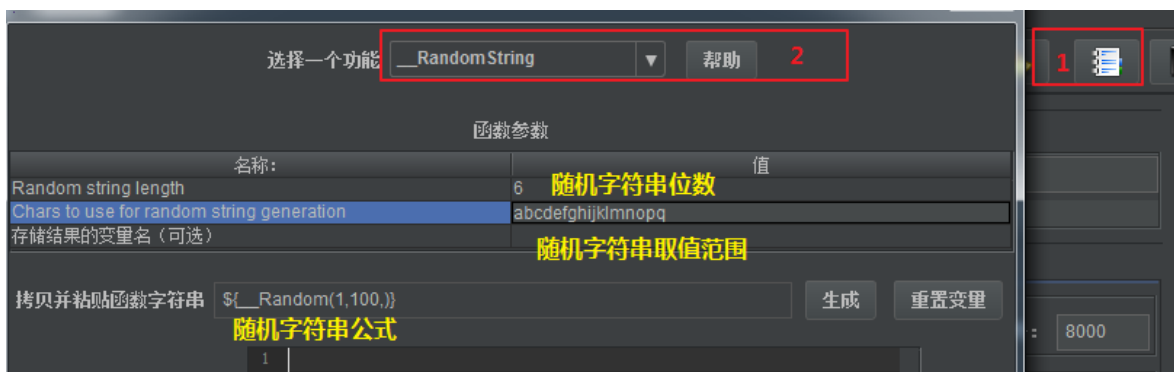
- 3、当同一个线程组里面有多个请求运用相同的参数的时候该如何处理



- 4、Jmeter如何读取CSV文件的数据
- 5、如何生成随机数



## 6、如何生成随机字符串



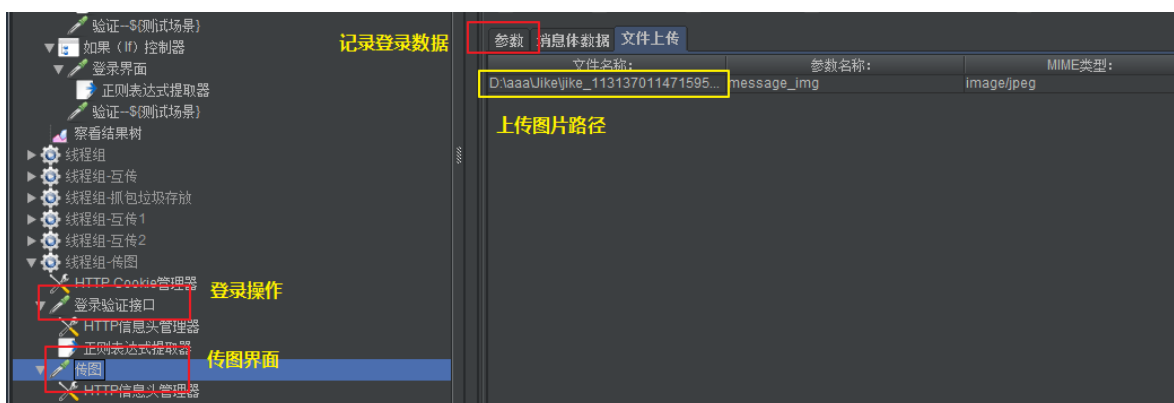
## 7、如果(if)控制器的使用方法



## 8、Jmeter如何连接数据库对数据库进行操作

## 9、如何跨线程组传递数据

## 10、在http请求里面如何上传图片



# 针对于性能测试的总结

## 性能测试

- 并发测试
- 请求的响应时间
- 错误率（不超过1/10000）
- 吞吐量
- tps()
- cpu()
- 内存()

在进行这几天的模拟面试过程中发现，性能测试是学生的薄弱环节，以及对接口的理解不够深入

在工作过程中进行的性能测试，可以说一些相对比较简单性能测试的东西

比如单接口的性能测试

在性能测试的过程中遇见的问题进行总结

## 1、内存泄露分为两种

现象：在进行比较长的时间运行或者是长时间的反复调用出现内存升高并不能得到及时的回收的情况

1、（app）在应用安装后，不进行任何的操作（只在后台运行），有内存泄露的情况（操作后内存会回收）

2、（服务器端）在某个接口进行长时间的运行后会内存泄露（引用的方法或者类没有关闭）（由于jdk的版本升级从1.7到1.8内存回收机制发生了改变，定位问题的时候应该考虑到相应的jdk的版本问题）

解决办法：增加内存回收的机制，在代码中正确的调用某些方法

## 2、慢查询

现象：当数据库的数据量比较大的时候，相应的响应时间会变长

发现慢查询的方法：先在数据库造少量的数据进行压测查看响应时间，然后对接口调用的相关表造大量的数据，如果响应的时间特别慢，则表明有慢查询的情况

解决办法：在数据库添加唯一索引

## 3、连接数比较小

现象：在并发到一定量的时候，接口请求的出错率会突然增加

基本上每一个应用都有连接数，具体是哪个应用连接数比较小要根据报错日志进行定位问题（如果是Tomcat将会报405，如果是数据库的问题在应用的log文件里查看报错信息进行定位问题）

解决办法：增大相应应用的连接数

## 4、代码优化

现象：即使数据库的数据比较小，相应的响应时间也比较长，tps比较小

- 1、查看有没有存在一个接口处理很多数据，多次调用外部依赖的品台，如果接口比较重的话，考虑拆分接口
- 2、查看有没有存在本能一次调用数据库能取出所有数据但是进行了多次调用数据库的情况
- 3、查看有没有使用的方法或者是类比较重的情况，如果有换成相对来说比较轻的类来实现该功能
- 4、如果需要用到的数据是每次调用都不会发生改变的数据，并且调用量比较大的话，可以考虑添加缓存机制（一般用非关系型数据库Redis进行保存）

## 5、其中并发测试也属于性能测试的范畴

并发测试存在于，存在两个不同的接口更改同一条数据的同一个字段（出现问题极难定位）

一个接口比如（加团，消费剩余金额，消费剩余商品，查询判断并更改某条数据的状态）

解决办法：应用Redis的原子性原理（单线程处理数据），增加Redis锁已解决并发的bug

运用更新数据库操作的原理如果数据已经更新，在对其进行更新将会更新不成功（包括找不到该条数据或更新的数据与本身存在的数据一致）

6、当软件部分优无可优的时候需要进行硬件的优化（比如：增加服务器的数量，增加网络的带宽，增加磁盘的性能（保证磁盘没有坏道，或者机械硬盘换固态硬盘），增加服务器的数量性能将是单点的性能乘服务器的数量。

7、正常情况下单点的TPS能够达到270，四个点的情况下支持日访问量为36W是没有问题的

8、具体性能测试关注的参数有：（90%请求的响应时间、错误率、吞吐量、cpu%、内存%）当tps小于200性能将得不到保障，出错率要小于千分之一，最好的体验接口的平均相应时间小于200毫秒体验较好，大于5秒将是无法忍受的时间

9、如果公司对单接口进行压测，进行压测的接口包括（应用的主接口，或者主逻辑接口，每个用户都要调用的接口，计算比较多的接口，在运算时返回数据的数据量比较大的接口，上传数据的接口，下载数据的接口）要测出接口的最优方案。

10、我测过的单点的TPS最大为一千三百多

## jmeter分布式压测

### 主电脑设置

1、\apache-jmeter-5.1.1\bin\jmeter.properties 修改remote\_hosts=127.0.0.1为从电脑ip（多个ip用英文逗号隔开），将server.rmi.ssl.disable=false 修改为server.rmi.ssl.disable=true，去掉注释，去掉mode=Standard注释；

2、菜单栏--运行--远程启动 配置从电脑

### 从电脑设置

1、jmeter--bin--jmeter.properties，将server.rmi.ssl.disable=false

修改为server.rmi.ssl.disable=true，去掉注释

2、jmeter--bin--jmeter-server.bat，运行

### 不显示jmeter界面的情况下去运行jmeter脚本

（linux同）

（jmeter--bin--shift右键--在此处打开命令窗口--dos运行）

jmeter -n -t C:\Users\lxy\Desktop\baidu.jmx -l lxy123.jtl

## 第九章（接口自动化）

### get和post的区别

1) get和post都是tcp连接并无差别，但是由于http的规定和服务器的限制，导致他们在应用过程中有一些不同

2) get参数通过url传递，post参数放在请求体中

3) get请求在url里面传递的参数是有长度限制的，post理论上来说没有长度限制，其限制的是服务器的处理能力

4) post对比get来说比较安全，因为get请求的参数暴露在url里面 所以不能用来传递敏感信息，如果post传递敏感信息在参数不加密的情况下也是不安全的

5) 对于get请求而言浏览器会把请求头和数据一起发送出去，而post请求是浏览器先发送请求头，服务器响应100，然后浏览器再发送请求数据。

## 测试用例框架

```
1  @classmethod
2  def setUpClass(cls):
3      print('开始之前执行一次')
4
5  @classmethod
6  def tearDownClass(cls):
7      print('结束之后执行一次')
8
9  def test_01(self):
10     print('这是test01')
11
12 def test_02(self):
13     print('这是test02')
```

### 单个用例的加入

```
1  #构造测试套件
2  suit = unittest.TestSuite()
3      #加入测试用例
4  suit.addTest(Test("test_01"))
5  suit.addTest(Test("test_02"))
6  #suit.addTest(Test("test_03"))
7      #执行测试
8  runner = unittest.TextTestRunner()
9  runner.run(suit)
```

### 全部加入

```
1  #构造测试套件
2  suit = unittest.TestSuite()
3  suit.addTest(unittest.makeSuite(Test))
4  runner = unittest.TextTestRunner()
5  runner.run(suit)
```

### 以列表的形式加入

```
1  suit = unittest.TestSuite()
2  test_cases = [Test('test_01'),Test('test_03')]
3  suit.addTests(test_cases)
4  runner = unittest.TextTestRunner()
5  runner.run(suit)
```

## 抓取接口

cookies设置为全局变量

```
1  res = requests.post(url=url,data=data)
2  self.assertIn('登录成功',res.text)
3  global cok
4  cok = res.cookies
```

## 登录--加购--查看购物车

```
1  # coding:utf-8
2  import unittest
3  import requests
4  class Api_AutoTest(unittest.TestCase):
5      def test_001(self):
6          url = "http://192.168.2.110/geeknet/user.php"
7          data = {
8              "username": "test001",
9              "password": "test001",
10             "act": "act_login",
11             "back_act": "./index.php",
12             "submit":""
13         }
14         res = requests.post(url=url,data=data)
15         self.assertIn('登录成功',res.text)
16         global cok
17         cok = res.cookies
18         def test_002(self):
19             url = "http://192.168.2.110/geeknet/flow.php?
step=add_to_cart"
20             data = {
21                 "goods": '{"quick":1,"spec":
[[],"goods_id":55,"number":"1","parent":0}]'
22             }
23             res = requests.post(url=url,data=data,cookies = cok)
24             def test_003(self):
25                 url = "http://192.168.2.110/geeknet/flow.php?
step=cart"
```

```

26 |         res = requests.get(url=url,cookies = cok)
27 |         self.assertIn("购物车 (1) ",res.text)
28 | if __name__== "__main__":
29 |     unittest.main()

```

try：监控这里的异常

except：异常处理代码

exception 为万能异常处理

except Exception as e:

```

1 | def test_04_subject_order(self):
2 |     url = "http://192.168.2.110/geeknet/flow.php?
step=checkout"
3 |     try:
4 |         res = requests.get(url=url,cookies=cok)
5 |         self.assertIn("应付款金额:",res.text)
6 |         print("subject order succeed")
7 |     except Exception as e:
8 |         print("subject order fail")

```

.yaml 文件 另一种标记语言 ( Yet Another Markup Language )

```

1 | -
2 |   url: http://192.168.2.110/geeknet/user.php
3 |   method: POST
4 |   data:
5 |     username: test026
6 |     password: test026
7 |     act: act_login
8 |     back_act: ./index.php
9 |     submit:
10 |   check:
11 |     - 登录成功

```

使用.yaml需要用ddt调用

```

1 | @ddt.file_data("D:\python\python\api_autotest\case\api.yaml")
2 | def test_01(self,**testdata):
3 |     url = testdata.get("url")
4 |     method = testdata.get("method")
5 |     data = testdata.get("data")
6 |     check = testdata.get("check")

```



```

1  -
2  url: http://192.168.2.110/geeknet/user.php
3  method: POST
4  data:
5      username: test026
6      password: test026
7      act: act_login
8      back_act: ./index.php
9      submit:
10 check:
11     - 登录成功

```

配置ini文件 ( config.ini )

```

1  [mysql]
2  host = 192.168.2.135
3  port = 3306
4  user = test026
5  passwd = test026
6  db_name = gk1
7  charset = utf8

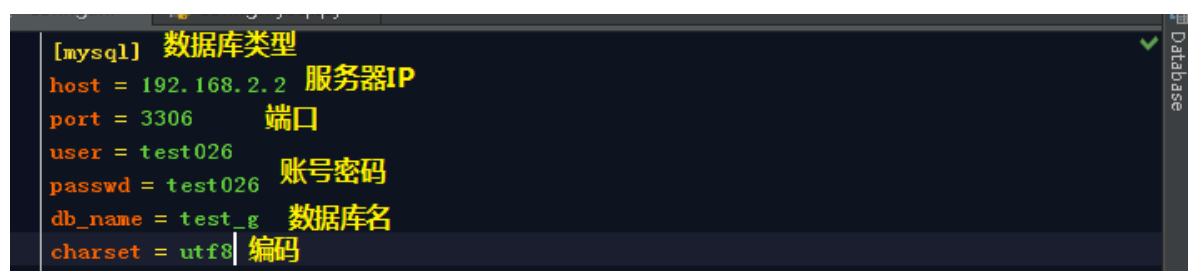
```

调用ini文件

```

1  config = configparser.ConfigParser()
2  config.read(r"D:\python\python\api_autotest\config\config.ini"
3  )

```



The screenshot shows a code editor with the following configuration and annotations:

```

[mysql] 数据库类型
host = 192.168.2.2 服务器IP
port = 3306 端口
user = test026
passwd = test026 账号密码
db_name = test_g 数据库名
charset = utf8 编码

```

```

1  if __name__ == "__main__":
2      toos = Test_mysql()
3      toos.conn_mysql()
4      sql = "select * from student2048"
5      a = toos.serch_sql(sql)
6      print(a)
7      toos.close_sql()

```

```

1  [mysql]
2  host = 192.168.2.110
3  port = 3306
4  user = root
5  passwd = 123456
6  db_name = gk1
7  charset = utf8
8
9  [email]
10 smtpserver = smtp.163.com
11 user = luokuo8080@163.com
12 password = luokuo1018
13 sender = luokuo8080@163.com
14 receive = okbug@foxmail.com

```

```

1  #coding:utf8
2  import pymysql
3  import configparser
4  config = configparser.ConfigParser()
5  config.read(r"D:\python\python\api_autotest\config\config.ini")
6  class Test_mysql():
7      def conn_mysql(self):
8          '''通过config读取配置，分别把config.ini里面的host, port,
9          user等读取出来方便后面连接数据库使用'''
10         host = config.get("mysql", "host")
11         port = config.get("mysql", "port")
12         user = config.get("mysql", "user")
13         passwd = config.get("mysql", "passwd")
14         db_name = config.get("mysql", "db_name")
15         charset = config.get("mysql", "charset")
16         self.connect = pymysql.connect(
17             #通过pysql构造：连接数据库的方法并把host, user, passwd等传进
18             去用来连接数据库
19             host=host,
20             port=int(port),
21             user=user,
22             passwd=passwd,
23             db=db_name,
24             charset=charset
25         )
26         #连接之后获取指针用以后面执行sql语句
27         self.cursor = self.connect.cursor()

```

```

26     def excute_sql(self, sql):
27         '''执行提交数据'''
28         self.conn_mysql() # 获取连接
29         self.cursor.execute(sql) # 用指针去执行sql
30         self.connect.commit()#提交sql
31     def serch_sql(self, sql):
32         """执行查询语句"""
33         self.conn_mysql() # 获取连接
34         self.cursor.execute(sql) # 用指针去执行sql
35         return self.cursor.fetchall() # 把执行结果通过fetchall
返回
36     def close_sql(self):
37         '''关闭数据库连接'''
38         self.cursor.close() # 关闭指针
39         self.connect.close() # 关闭数据库连接
40 if __name__ == "__main__":
41     toos = Test_mysql()
42     toos.conn_mysql()
43     sql = "select * from student2048"
44     a = toos.serch_sql(sql)
45     print(a)
46     toos.close_sql()
47     # sql = 'select * from student2048'
48     # conf_mysql = Test_mysql()
49     # conf_mysql.conn_mysql()
50     # a = conf_mysql.serch_sql(sql)
51     # print(a)
52     # conf_mysql.close_sql()

```

```

1  #coding:utf8
2  from email.mime.text import MIMEText
3  import configparser
4  import smtplib
5  import os
6  config = configparser.ConfigParser()
7  config.read('D:\\python\\python\\api_autotest\\config\\config
.ini')
8  class send_youjian():
9      def send_mail(latest_report):
10         f = open(latest_report,'rb')
11         body = f.read()
12         f.close()
13         smtpserver = config.get('email' , "smtpserver")

```

```

14     user = config.get("email", "user")
15     password = config.get("email", "password")
16     sender = config.get("email", "sender")
17     receive = config.get("email", "receive")
18     subject = "API自动化测试报告--ZhangZhiyun"
19     # 邮件标题
20     msg = MIMEText(body, "html", "utf-8")
21     # 把正文转化成HTML格式
22     msg["from"] = sender
23     # 从哪发送（获取上文的发送者）
24     msg["to"] = receive
25     # 发送到哪（获取上文的接收者）
26     msg["subject"] = subject
27     # 加上标题
28     smtp = smtplib.SMTP()
29     # 实例化一个连接对象
30     smtp.connect(smtpserver)
31     # 跟163邮箱建立连接
32     smtp.login(sender, password)
33     # 用账号和授权码登录邮箱
34     smtp.sendmail(sender, receive, msg.as_string())
35     # 发送邮件
36     smtp.quit()
37     def last_report(report_dir):
38         lists = os.listdir(report_dir) # 获取report_dir地址下
    的所有文件并以列表形式展示
39         lists.sort(key=lambda fn: os.path.getmtime(report_dir
    + "\\\" + fn))
40         #将列表从小到大排序 os.path.getmtime()获取最近修改的时间
41         file = os.path.join(report_dir, lists[-1])
42         return file
43     if __name__ == "__main__":
44         report_dir = 'D:\\python\\python\\api_autotest\\report'
45         mail = Send_youjian
46         latest_report = mail.last_report(report_dir)
47         mail.send_mail(latest_report)

```

## Jenkins

Jenkins是一款持续集成工具（CI持续集成）

Jenkins

通过关联svn（git）（代码仓库）

以及maven（对代码进行编译）

部署到测试环境

Jenkins如何部署：

首先开发修复完代码之后会把最新代码放到指定的svn代码仓库，然后我们找到对应的应用点击立即构建，Jenkins就会自动拉取代码仓库的最新代码，然后用maven进行打包，然后再部署到Tomcat的webapps下面，然后重启应用部署成功。

主干（trunk）--树干

分支（branch）--树枝

测试一般在分支上面测试测试完毕之后需要在分支上面进行回归测试，然后没问题了让开发把代码合并到主干，然后再进行回归测试，没问题再发布。

```
1 | https://192.168.2.2:8443/svn/YunChuangRepository/trunk      （主  
   | 干）  
2 | https://192.168.2.2:8443/svn/YunChuangRepository/branch/个人中  
   | 心 （分支）
```

jenkins如何回滚？

首先我们先获取到要回滚的版本号，（每个应用每构建一次就会产生一个版本号），然后用参数进行构建，把版本号放到参数里面点击确定就会回滚到指定的版本。

如何搭建jenkins？（部署在服务器上，centos7操作系统）

- 首先我们先下载一个jenkins的war包（jenkins.war）
- 然后把jenkins的war包放到Tomcat的webapps里面
- 再去浏览器对应的页面打开，按照上方引导安装



# 公司概况与测试用例

---

F		G	
不通过	1.有一定接口测试功底；2.没有bug定位能力 3.有一定特殊情况的处理能力；4.有测试立场		
不通过	完全很不耐烦的样子； 全程没抬头 工作中觉得自己没有什么缺点		
不通过	力东：优点：测试基础不错，风险与测试目的较清楚，有一定接口及自 化经验 不足：工作热情上可以表现再主动、积极 兰兰：学习上需加强，等价类概念模糊，生产问题处理优先级妥当 但优化流程缺失		
不通过	1.问题定位能力一般 2.接口测试能力一般 3.态度较好 4.没有风险意识		
通过	文言：1.测试用例设计思路清晰；2.排查问题思路清晰；3.有一定的接口 测试功底；4.测试流程一般 兰兰：人不错，可以用，缺乏学习		
不通过	兰兰：1.流程尚可；2.需求评审不错 缺点：1.对于争议问题和开发私下解决，没有立场；2.不能独立测试，对 于关键干系人不能分析清楚 3.上线问题处理能力偏弱；4.学习能力较弱		
不通过	候选人表示不想做笔试		
不通过	1.流程不规范；2.没有测试立场，企图和开发搞好；3.接口测试能力很一 般 4.没有独立思考的能力； 5.表达能力很弱		
不通过	测试用例设计不全面		
不通过	1.问题定位能力混乱； 2.态度较好 3.自己说自己接口能力不错，实际一般		



---

与风险意识不错，接口、自动化都有具体实践；

F	G
018/12/06 QA集中面试反馈表	
面试情况	面试评语
不通过	测试基础不良，但有学习空间，思路太窄，需要有人很耐心地带在身边
不通过	不足：测试工作不够独立，沟通太内向
不通过	兰兰：1.用例设计偏弱；2.无法独立完成项目测试工作； 3.推动问题能力较弱
不通过	了解东西都没有深入
--	
不通过	需求无法理解，学习能力有，不学习
不通过	1.沟通能力弱，理解能力弱 2.问题定位能力弱 3.接口测试仅仅是了解
不通过	兰兰：1.测试流程不规范；2.用例设计不能把握核心；3.不太爱学习
不通过	力东：优 测试经验与风险意识不错，接口、自动化都有具体实践； 隐患：工作稳定性 文言：不建议录用；1.不是很适合目前的团队，有很多不切实际的想法和期许 2.可能工作上过于有主见，不好配合
不通过	测试用例设计较差
不通过	1.没有bug定位能力；2.自言自语了很久 3.设计用例可能会考虑欠缺，考虑不周全，不是一个有立场的测试，不建议录用
--	
不通过	兰兰：1.用例设计偏弱；2.无法独立完成项目测试工作； 3.推动问题能力较弱
--	
通过	兰兰：表达能力OK，风险把握OK 缺点：1.测试基础偏弱； 2.对于问题处理推动能力偏弱，但知道会把握第三方资源 3.业务把握有待精进；

---

# 公司测试概况

---

## 1、需求什么时候进入迭代

?

## 2、迭代时间是怎么进行安排的

第一天（ ）上午开计划会：明确迭代做哪些东西，评估任务量

第二天编写测试用例

第三四天上午将计划用例评审了，之后进行测试，

到最后两天（九、十），回归测试跟上线

第十天下午开回顾会（吐槽，反馈，批判，改进问题）（开发测试人员）

sqa（监管人员，质量监督人员）监督迭代

## 3、迭代的人员安排

开发：测试 = 3：1（大概）前端与测试数量差不多

## 4、测试分几轮，测试人员是怎么进行分配的

SIT一页一页测

sit回归测试

uat回归测试

uat环境进行交叉测试（大项目）

生产环境回归测试

## 5、一天能写多少条测试用例

七八十条（累）四五十条（正常）

### 5.1一天执行多少测试用例

二三十条（测试）七八十（回归阶段）

## 6、一个项目能写多少测试用例

几千（大）几百（小）两百（小）

## 7、一个项目多少人测试

二到三人

## 8、多久迭代一次

一周 十天（两星期）

## 9、一天能发现多少bug

看开发 七八个

## 10、一个项目能发现多少bug

有多又少，一百多个（一般）

## 11、公司都有哪些环境

sit、uat、生产环境、（开发人员的环境）

## 12、公司的小组如何划分

包含测试、前端、后端

# 测试用例

---

### 小程序测试点

小程序的测试我们除了要考虑最基本的功能测试，接口测试和页面UI样式测试，业务流程测试和数据流测试外有自己的一些特有的测试

首先小程序有自己的一些特点：

1）分为三种版本类型：开发版，体验版，正式版

开发版和体验版无需审核，通过扫小程序二维码才能访问

开发版和体验版的区别在于，在开发版小程序二维码有效期比较短。

正式版需要通过微信审核流程。

#### 1.权限测试

用户首次访问小程序，主动获取微信授权，通过获取openid，生成转uid，并存储昵称、头像等信息。后续用户若杀掉小程序重新进入，无需重新授权。

1）未授权微信登录小程序

未授权时会首先弹出权限提醒，授权之后才可访问小程序

## 2) 已授权微信登录小程序

授权微信访问小程序，意味着自己的微信账号可被小程序管理方所获取，自动以微信的身份行使业务操作权限，比如咨询、支付、数据查询等

## 3) 同一微信号在不同手机端登录授权查看数据权限

## 4) 重新授权

取消权限之后再次重新授权

## 2.兼容性测试

### 1) 手机操作系统

常规的手机端OS为：Android ( 8.x/7.x/6... )、IOS ( 11.x/10.x/9.x... )

### 2) 微信版本测试

### 3) 小程序码的兼容性测试

因为目前小程序不支持直接分享朋友圈，只能分享微信好友。

### 4) 屏幕兼容性测试

比如4.0/4.7/5.0/5.7等等屏幕尺寸

## 3、网络测试

### 1) 在wifi和2G/3G/4G网络下是否可以正常访问

### 2) 网络切换的测试比如从2G/3G/4G切换为wifi

### 3) 无网络测试以及从有网到无网测试

## 6、更新测试

更新有2种机制

### 1) 未启动时更新（后台静默更新到最新版本）

2) 启动时更新（每次冷启动都会检查是否有最新版本，如果有就异步下载更新包，并先使用老的包进行启动，新版本会在下次冷启动时应用）

## 7、页面样式测试

查看页面样式/ui交互/导航栏/轮播图/背景图/字体等测试

同一公司下多个公众号之间需要用户帐号互通，微信开放平台提供了UnionID机制

在不同的公众账号下OpenID是不一样的，而他们的UnionID却是一样的

UnionId和uid是对应关系；OpenId和sourceId是对应关系。

## H5的测试点

- 1 功能、性能、接口、UI
- 2 登录模块、翻页返回和刷新功能、屏幕旋转、网络切换和弱网测试、链接分享测试、兼容性测试、加载速度测试

一般除了最基本的功能，接口，性能和UI测试之外也有自己的一些特有的测试

### 1.针对登录模块

- ①登录app那么进入H5也应该是登录状态
- ②若app未登录点击对应的按钮应该触发单点登录提示登陆，以及取消之后再次点击是否提示登录
- ③页面session或cookie过期是否提示重新登录

### 2.翻页/返回和刷新

- ①在一页和多页的情况下数据展示是否正常能否翻页跳页
- ②翻页之后页面的焦点是否定位正确
- ③点击页面内的返回按钮是否符合业务逻辑
- ④点击物理返回按钮是否符合业务
- ⑤单次刷新能否正常。多次连续下滑刷新是否正常

### 3.屏幕旋转

### 4.网络切换弱网测试

在2G/3G/4G和WiFi情况下 是否能够正常访问

以及2G/3G/4G和WiFi的切换情况下是否能够正常

### 5.滑动定位是否准确

手指滑动是否流畅，手指点击时焦点是否定位正确

### 6.H5链接分享测试

- ①链接分享标题是否正确
- ②分享之后数据/样式是否正确合理
- ③二次分享以及分享之后是否可以用户操作
- ④以及点赞评论浏览量测试

## 7.兼容性测试

- ①不同操作系统兼容性比如安卓4.0/5.0/6.0和ios 8-10-12
- ②不同手机型号的兼容性比如小米/三星/华为/魅族 和苹果6-7-8-x
- ③不同分辨率比如720/1080/2k屏幕
- ④手机浏览器包括自带的浏览器/chrome浏览器/UC浏览器等

## 8.加载速度

- ①不同网络下的加载速度
- ②首次打开的加载速度，没有缓存和清理缓存之后的加载速度
- ③加载资源的处理比如图片压缩或者放到CDN（内容分发缓存服务器）

## 优惠券

- 1.购买时使用优惠券能否正常结算
- 2.使用过期的优惠券
- 3.不同等级的用户是否可以使用优惠券
- 5.是否可以叠加使用优惠券
- 4.优惠券支付之后是否会失效
- 6.订单支付之后优惠券是否会退换（一般不退）
- 7.使用优惠券支付之后产生退款，退款金额是否正常
- 8.支付失败之后优惠券是否再次可用
- 9.支付过程中取消支付优惠券是否还在
- 10.优惠券是否针对特定商品
- 12.调用第三方支付是否可以正常支付
- 14.支付时网络不好，重复点击是否会产生二次支付（测接口幂等）
- 13.支付之后是否可以正常跳转，订单状态是否可以正常更新
- 15.订单支付之后产生退货，退款流程校验（一般服务购买之后不可退货）
- 16.购买服务之后服务是否会生效
- 11.数据库测试，待支付/服务中/已完成/退款中的节点状态是否与系统一直是与数据库里面的一致
- 17.兼容性测试，在不同手机，不同系统，不同分辨率下的兼容性测试

18.接口的测试，待支付/服务中/已完成/退款中支付（包括抓包越过验证修改参数，幂等，参数组合，业务逻辑等等）

## 纸杯

功能方面：

1.是否能盛水

2.能盛多少水

性能：

1.杯子是否容易变形

2.杯底设计是否稳固不易倾斜

7.是否易被风吹倒

.能够叠加多少个纸杯

3.是否能重复使用，能重复使用多少次

4.能盛水的温度

6.能否保温

8.是否有隔热性能

安全：

1.杯口是否圆润

2.材质是否安全

长时间放置是否会有水溶物质

4.长时间放置是否漏水

3.是否易燃

兼容性：

1.能否盛固体，酒精，碳酸饮料，腐蚀性液体等

2.是否可以叠加

易用性：

1.是否便于携带

5.杯子材质是否轻盈

杯口杯底大小是否设计合理



是否方便握持

2.是否易于老人小孩使用

6.是否容易变形

7.掉落是否容易破损

界面：

1.界面图案设计是否合理

2.图案是否容易掉色

3.杯子尺寸形状厚度图案设计是否符合需求

## **笔**

功能测试

1.能否写字

2.能否悬挂

3.是否配带说明书

性能测试：

1.字迹粗细

2.长时间放置是否能正常书写

3.出墨是否连贯

4.能书写多长时间

5.书写之后墨水的风干时间

6.能否在极寒或者高温条件下正常书写

7.笔帽能够开合多少次

8.笔珠是否容易脱落

9.字迹能够保留多久

兼容性测试：

1.是否兼容不同笔芯

2.是否可以在不同纸张上面书写

3.配件是否可以更换

可靠性测试：

- 1.笔的元器件是否结合牢固
- 2.笔的螺纹是否容易滑丝
- 3.橡胶圈是否容易老化
- 4.笔芯是否容易漏墨
- 5.笔是否可以拆卸安装
- 6.笔身颜色是否容易掉色

压力测试：

- 1.从多高的高度容易摔坏
- 2.笔杆笔尖能承受多大的压力

易用性测试：

- 1.笔是否适合老人小孩使用
- 2.笔杆是否方便握持
- 3.笔杆是否有防滑措施
- 4.是否易于放置

安全测试：

- 1.笔是否尖锐容易伤人
- 2.笔杆/墨水是否对人体有害

文档测试：说明书内容是否完整正确

界面测试：1.是否与设计稿一致

## 二维码

二维码的原理：二维码其实里面是一个加密的URL，是通过QR码的形式进行编码，在二维码里面黑色的像素块代表1，白色的像素块代表0，然后扫描的时候其实是识别的二维码里面的像素块然后通过解码的方式转换成对应的URL。

功能测试

- 1.二维码是否可以正常被识别
- 2.自己是否可以扫描自己的二维码
- 3.截图或拍照的二维码是否可以被正常识别

## 兼容性测试

- 4.不同软件的二维码是否可以正常扫描
- 5.前置摄像头是否可以正常识别
- 6.二维码不清晰是否可以正常识别
- 7.二维码在不同的材质上面是否可正常识别

## 性能测试

- 8.可以被识别的最近/最远距离
- 9.可被识别的最大或最小尺寸
- 10.扫描方向/角度校验
- 11.二维码破损程度扫描校验
- 12.可同时被多少人扫描
- 13.扫描框里面放置多个不同二维码扫描校验

## 可靠性测试

- 14.是否有扫描次数上限
- 15.二维码是否有有效期
- 16.失效的二维码是否可被识别

## 压力测试

- 17.被多次转发的二维码是否可被识别
- 18.二维码有透明阻挡物是否可正常识别
- 19.修改二维码颜色是否可以正常扫描
- 20.手动修改二维码是否可以识别
- 21.在阴暗或者高亮的情况下扫描能否正常识别

## ATM机

### 插卡部分：

- 1.插卡方向
- 2.不插卡
- 3.无效卡例如银行卡消磁，非银行卡，会员卡，

密码部分：

- 1.输入正确密码
- 2.插卡不输入密码
- 3.输入错误的6位密码
- 4.输入5位或7位密码
- 5.密码输错3次是否吞卡
- 6.密码输入错误次数是否累计
- 7.更改密码之后输入老密码
- 8.如果一段时间不操作是否会退卡

交易部分：

- 1.是否可以正常取款
- 2.非100整数是否可以取款
- 3.取款金额大于卡内余额
- 4.取款金额大于每日限额
- 5.不输入取款金额
- 6.取款金额输入0
- 7.取款之后卡内余额是否扣减正确
- 8.取款机余额不足
- 9.取款金额与出钞金额是否一致
- 10.出钞之后没有取钞
- 11.单日取款次数
- 12.单笔最高取款金额
- 13.单日取款最大金额
- 14.跨行取款

退卡部分

- 1.是否能够正常退卡
- 2.正常打印凭条
- 3.凭条信息是否与取款信息匹配

4.取卡之后是否自动退出系统

5.退卡之后不取卡是否会提示

## 抢红包

1.首先是否能够正常发红包

5.能否正常抢红包

3.无网络/网络差的情况下是否能够发/抢红包

13.余额不足是否能够发送红包

10.红包金额为0时能否发送

10.红包金额为空时能否发送

14.红包金额是否支持小数

9.红包金额与设置金额是否一致

6.红包金额最大和最小校验

2.红包金额是否能随机

4.发送红包之后余额扣减是否正确

22.红包支付方式校验

11.抢到红包后能否正常存入余额

7.单次红包数量上限校验

8.发送红包之后红包数量是否设置一致

16.自己发红包自己领取校验

15.单个人是否可以二次领取红包

17.抢完红包之后手气最佳校验

19.抢红包记录是否与每个人所抢金额一致

21.红包历史排序是否按照抢红包的先后顺序排序

18.以抢红包和未抢红包的金额数量是否计算正确

20.红包抢完时间计时是否正确

12.延期未领取或未领取完的红包是否会在规定的时间内退回

## 添加购物车

## 1.界面测试

1. 界面布局、排版是否合理；文字是否显示清晰；不同卖家的商品是否区分明显。

## 2.功能测试

### 未登录时：

1. 将商品加入购物车，页面跳转到登录页面，登录成功后购物车里数量增加；
2. 点击购物车菜单，页面跳转到登录页面。

### 登录后：

1. 所有链接是否跳转正确；
2. 商品是否可以成功加入购物车；
3. 购物车商品总数是否有限制；
4. 商品总数是否正确；
5. 全选功能是否好用；
6. 删除功能是否好用；
7. 填写委托单功能是否好用；
8. 委托单中填写的价格是否正确显示；
9. 价格总计是否正确；
10. 商品文字太长时是否显示完整；
11. 店铺名字太长时是否显示完整；
12. 创新券商品是否打标；
13. 购物车中下架的商品是否有特殊标识；
14. 新加入购物车商品排序（添加购物车中存在店铺的商品和购物车中不存在店铺的商品）；
15. 商品删除后购物车商品总数是否减少；
16. 购物车结算功能是否好用。

## 3.兼容性测试

不同浏览器测试。

## 4.易用性测试

删除功能是否有提示；是否有回到顶部的功能；商品过多时结算按钮是否可以浮动显示。

## 5.性能测试

压力测试；并发测试。

## 购物流程

- 1、未注册购物
- 2、注册后购物
- 3、填写注册信息校验
- 4、购物车页面校验
- 5、结算校验

# 零碎

---

## 知识点

---

### 一、http请求有哪些方法（8种）常用的有哪些？

- OPTIONS ( options )
- HEAD ( head )
- PUT ( put )
- TRACE ( trace )
- CONNECT ( connect )
- 【GET】向特定的资源发出请求。
- 【POST】向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。
- 数据被包含在请求体中。POST请求可能会导致新的资源的创建和/或已有资源的修改。
- 【DELETE】请求服务器删除Request-URI所标识的资源。

### 二、http状态码代表的意思？

（404、100、200、300、400、500、1开头、2开头、3开头、4开头、5开头）

404：（未找到）服务器找不到请求的网页。

502：（错误网关）服务器作为网关或代理，从上游服务器收到无效响应。

100：（继续）请求者应当继续提出请求，服务器返回此代码表示已经接收到请求的第一部分，正在等待其余部分。

200：（成功）服务器已成功处理了请求。通常，这表示服务器提供了请求的网页。

300：（多种选择）针对请求，服务器可执行多种操作。服务器可根据请求者（user agent）选择一项操作，或提供操作列表供请求者选择。

400：（错误请求）服务器不理解请求的语法。

500（服务器内部错误）服务器遇到错误，无法完成请求。

http状态返回代码

- 1XX: 信息，服务器收到请求，需要请求者继续执行操作。
- 2XX：成功，操作被成功接收并处理。
- 3XX：重定向，需要进一步的操作以完成请求。
- 4XX：客户端错误，请求包含语法错误或无法完成请求。
- 5XX：服务器错误，服务器在处理请求的过程中发成了错误。

### 三、app测试的测试点？

- 1 功能测试、接口测试、性能测试、UI测试、
- 2 兼容性测试、渠道包测试、
- 3 安装和卸载测试、升级和更新测试、
- 4 安全测试、权限测试、网络测试、中断测试、
- 5 消息推送测试、手势操作测试、软件占用系统资源测试

### 四、定位问题的4种方法？

- http状态码
- 抓包
- 查看动态日志
- 数据库

### 五、web测试与app测试的区别？

- 1 web测试和APP测试测试流程大致相同，不同点在于web项目基于B/S架构，而APP基于C/S架构，所以系统测试时会有细节上的不同。
- 2 从系统架构上来说，web测试只是更新了服务器端，客户端就会同步更新，能保证每个用户的客户端完全一致；而APP端如果修改了服务器端的资源或者系统架构，客户端所使用的所有核心版本都需要回归测试；
- 3 兼容性方面，web测试是基于浏览器的，测试时一般考虑不同内核的浏览器；而APP测试考虑的是手机型号与系统，屏幕尺寸与分辨率，还有其他APP特有的测试，比如权限测试、安装卸载测试、升级更新测试、手势操作测试等；总的来说，测试产品千变万化但测试的思维是不变的。

首先web测试跟app测试在测试流程上没有什么区别，我个人觉得是由于载体不一样，所以在系统测试时会有一些细节不一样，Web项目一般都是B/S架构的，是基于浏览器的，而app是C/S架构，必须有客户端。

**从系统架构上面来说，**

web测试只是更新了服务器端，客户端就会同步更新 而APP端如果修改了服务器端的资源或者系统架构，严格来说客户端所使用的所有核心版本都需要回归测试一下



## 然后从兼容性方面

首先web是基于浏览器的，一般测试的构成中以浏览器的兼容性为主，我们一般考虑是不同内核的浏览器，比如说IE，谷歌，火狐等浏览器 而APP的兼容性必须依赖手机或pad，我们在测试的过程中，不仅要考虑不同的平台，主要就是安卓和iOS，而且要考虑主流的机型，其次是不同的手机型号，系统版本，屏幕尺寸和分辨率等，相比较web测试APP测试又多了一些专项测试比如权限测试，安装卸载，升级更新，中断 网络手势 和消息推送以及占用系统资源的测试 总的来说，测试的产品千变万化但测试的思维是不变的

## 六、如果网页打不开如何排查问题？

- 1 无网络、网络差请求超时、无网络权限、网址错误、请求中数据丢失
- 2 浏览器故障、与浏览器不兼容、浏览器设置了代理
- 3 服务器拒绝、服务器不存在请求资源、服务器不支持的协议、服务器拥堵
- 4 本地进程占用过高、Nginx/DNS/网站证书出现问题、开启了防火墙、host配置问题

1、没有网（检查网线，检查网络是否正常）

2、浏览器有故障

3、网址是否正确

4、服务器是否拒绝请求

5、检查防火墙是否打开

6、检查是否配置了host

7、浏览器是否授权使用网络

8、浏览器版本与网页不兼容

9、网络比较差导致请求超时

10、服务器不存在请求的资源

11、服务器不支持当前的http协议

12、请求中数据丢失

13、服务器出现拥堵

14、浏览器设置了代理

15、本地的资源占用过多

16、Nginx出现问题

17、Dns出现了问题

## 七、http与https的区别？

http：端口号 80（超文本传输协议）特点：协议简单 灵活，允许传输任意类型的数据。协议每次连接只处理一个请求。

https：端口号 443（超文本加密传输协议）（SSL+HTTP）要比http安全。

## 八、小程序测试点？

首先小程序有三种版本类型：开发版，体验版，正式版

开发版和体验版无需审核，通过扫小程序二维码才能访问

开发版和体验版的区别在于，开发版小程序二维码有效期比较短。

正式版需要通过微信审核流程。

小程序的测试我们除了要考虑最基本的

- 1 功能测试
- 2 接口测试
- 3 UI测试
- 4 业务流测试
- 5 数据流测试

外有自己的一些特有的测试

- 1 权限测试
- 2 兼容性测试
- 3 网络测试
- 4 更新测试

### 1、权限测试

用户首次访问小程序，主动获取微信授权，通过获取openid，生成uid，并存储昵称、头像等信息。后续用户若杀掉小程序重新进入，无需重新授权。

#### 1) 未授权微信登录小程序

未授权时会首先弹出权限提醒，授权之后才可访问小程序

#### 2) 已授权微信登录小程序

授权微信访问小程序，意味着自己的微信账号可被小程序管理方所获取，自动以微信的身份行使业务操作权限，比如咨询、支付、数据查询等

3) 同一微信号在不同手机端登录授权查看数据权限

4) 重新授权

取消权限之后再次重新授权

## 2、兼容性测试

1) 手机操作系统

常规的手机端OS为：Android ( 8.x/7.x/6... )、IOS ( 11.x/10.x/9.x... )

2) 微信版本测试

3) 小程序码的兼容性测试

因为目前小程序不支持直接分享朋友圈，只能分享微信好友。

4) 屏幕兼容性测试

比如4.0/4.7/5.0/5.7等等屏幕尺寸

## 3、网络测试

1) 在wifi和2G/3G/4G网络下是否可以正常访问

2) 网络切换的测试比如从2G/3G/4G切换为wifi

3) 无网络测试以及从有网到无网测试

## 4、更新测试

更新有2种机制

1) 未启动时更新 ( 后台静默更新到最新版本 )

2) 启动时更新 ( 每次冷启动都会检查是否有最新版本，如果有就异步下载更新包，并先使用老的包进行启动，新版本会在下次冷启动时应用 )

## 5、页面样式测试

查看页面样式/ui交互/导航栏/轮播图/背景图/字体等测试

同一公司下多个公众号之间需要用户帐号互通，微信开放平台提供了UnionID机制

在不同的公众账号下OpenID是不一样的，而他们的UnionID却是一样的

UnionId和uid是对应关系；OpenId和sourceId是对应关系。

## 九、H5测试点

一般除了最基本的

1 | 功能，接口，性能和UI测试

之外也有自己的一些特有的测试

## 1.针对登录模块

- ①登录app那么进入H5也应该是登录状态
- ②若app未登录点击对应的按钮应该触发单点登录提示登陆，以及取消之后再次点击是否提示登录
- ③页面session或cookie过期是否提示重新登录

## 2.翻页/返回和刷新

- ①在一页和多页的情况下数据展示是否正常能否翻页跳页
- ②翻页之后页面的焦点是否定位正确
- ③点击页面内的返回按钮返回逻辑是否符合业务
- ④点击物理返回按钮是否符合业务
- ⑤单次刷新能否正常。多次连续下滑刷新是否正常

## 3.屏幕旋转

## 4.网络切换弱网测试

在2G/3G/4G和WiFi情况下 是否能够正常访问  
以及2G/3G/4G和WiFi的切换情况下是否能够正常

## 5.滑动定位是否准确

手指滑动是否流畅，手指点击时焦点是否定位正确

## 6.H5链接分享测试

- ①链接分享标题是否正确
- ②分享之后数据/样式是否正确合理
- ③二次分享以及分享之后是否可以用户操作
- ④以及点赞评论浏览量测试

## 7.兼容性测试

- ①不同操作系统兼容性比如安卓4.0/5.0/6.0和ios 8-10-12

②不同手机型号的兼容性比如小米/三星/华为/魅族 和苹果6-7-8-x

③不同分辨率比如720/1080/2k屏幕

④手机浏览器包括自带的浏览器/chrome浏览器/UC浏览器等

## 8.加载速度

①不同网络下的加载速度

②首次打开的加载速度，没有缓存和清理缓存之后的加载速度

③加载资源的处理比如图片压缩或者放到CDN（内容分发缓存服务器）

## 十、OSI的七层模型、tcp/ip五层模型（知道物理层，传输层，应用层有哪些东西？）

（Open System Interconnect开放式系统互联，）

osi七层	tcp/ip五层	
应用层		
表示层		
会话层	应用层	HTTP FTP TFTP DNS SMTP Telnet
传输层	传输层	tcp udp
网络层	网络层	IP ICMP RIP IGMP
数据链路层	数据链路层	ARP RARP IEEE802.3 PPP CSMA/CD
物理层	物理层	FE自协商 Manchester MLT -3 4A PAMS

就好比：修路 架桥 地图 汽车 货物

应用层

网络服务与最终用户的一个接口

协议有：HTTP FTP TFTP SMTP DNS Telnet HTTPS POP3 DHCP

传输层

定义传输数据的协议端口号，以及流控和差错校验。

协议有：TCP UDP,数据包一旦离开网卡即进入网络传输层

## 十一、cookie与session的区别？

- 1.cookie的数据是存放在客户端的浏览器上，session数据存放在服务器上
- 2.cookie不是很安全，别人可以分析存放在本地的cookie进行cookie欺骗
- 3.Session会在一定的时间内存放在服务器上面，当访问增多的时候，会比较占用服务器的资源。
- 4.一般情况下 单个的cookie大小不能超过4k，很多浏览器限制一个web站点最多保存20个cookie，一般一个浏览器保存最多300个cookie。
- 5.一般来说，将登录信息账号密码以及隐私信息等重要信息存放在Session 里面。其他信息如果需要保留，可以存放cookie 里面。

## 十二、什么是token？

- 1 Token是服务端生成的一串字符串，以作客户端进行请求的一个令牌，当第一次登录后，服务器生成一个Token便将此Token返回给客户端，以后客户端只需带上这个Token前来请求数据即可，无需再次带上用户名和密码。

Token(令牌)：一般应用于app验证

最简单的token是由：uid(用户id)、time(当前时间的时间戳)、sign(签名)，可以把不变的一些参数放在token里面方便查询。

## 十三、tcp、udp的区别？

- 1 tcp是一种可靠的传输协议，与服务器交互需要三次握手四次挥手，而且它有确认、重传、拥堵等机制，在带来稳定性的同时也占用了系统资源，比较耗时；而udp的话，是无状态的传输协议，没有各种机制，所以传输很快，但又不可靠不稳定，网络不好可能会丢包。

Tcp 是一种可靠的传输协议，是面向连接的，建立连接时需要三次握手，断开时候需要四次挥手，所以tcp是比较稳定的，它规定浏览器向服务器发送数据时服务器必须发回确认，但是它也有缺点，因为tcp在与服务器进行交互的过程中，其中包含了确认机制，重传机制，拥堵机制，等等相当于Udp 来说比较耗时间，而且每次连接都会占用系统资源。

而Udp协议没有tcp的握手，确认，重传，拥堵等机制，它是一个无状态的传输协议，所以它在传输的过程中会非常的快，但是同时它又是不可靠的不稳定的，比如说在和服务器进行交互的过程中如果遇到网络状态不好的情况会造成数据包的丢失。

## 十四、什么是三次握手，什么是四次挥手？（用自己的话说出来）

三次握手：

第一次：主机A发送建立连接的数据包到服务器，主机B接受，A要求建立联机

第二次：主机B收到请求后确认联机信息，向A发送建立连接的确认包

第三次：主机A收到之后检查确认包是否正确，若正确，主机A再发送确认包，主机B收到并确认，然后建立连接成功

四次挥手：

第一次：客户端A发送一个结束包，用来关闭A到B的数据传输，客户端进入结束等待状态

第二次：服务器B收到这个结束包，并发回一个确认包，服务器进入等待关闭状态

第三次：服务器B在发送一个结束包用来关闭数据传输，服务器进入最后确认状态

第四次：客户端A收到结束包之后向B发送一个确认包且序号+1 服务器B收到之后关闭连接，A等待2MSL 之后A也关闭连接

2MSL意思是2倍等待时间。

## 十五、什么是对称性加密？什么是非对称性加密？（区别）知道MD5

对称性加密（AES加密，DES加密）

加密解密都使用相同密钥

非对称性加密(RSA加密)

加密和解密都使用不同密钥（私钥可以推测出公钥，公钥无法推测出私钥）

MD5

不可逆，压缩性

- |    |   |
|----|---|
| 1  | <b>BASE64:</b>  |
| 2  | 编码方式（8位字节代码），二进制与字符串相互转换                                |
| 3  | <b>MD5:</b>   |
| 4  | <b>Message Algorithm</b> （消息摘要算法第五版），散列函数（哈希算法）_不可逆，压缩性 |
| 5  | <b>DES:</b>   |
| 6  | <b>Data Encryption Standard</b> （数据加密标准），对应算法是DEA       |
| 7  | 特点：1. 对称加密 2. 同一个SK                                     |
| 8  | <b>AES:</b>   |
| 9  | <b>Advanced Encryption Standard</b> （高级加密标准）            |
| 10 | 特点：1. 对称加密 2. 一个SK扩展成多个子SK，轮加密                          |
| 11 | <b>RSA:</b>   |
| 12 | 特点：1. 非对称加密，即：PK与SK不是同一个                                |
| 13 | 2. PK用于加密，SK用于解密  |

- |    |  |
|----|--|
| 14 | 3. PK决定SK，但是PK很难算出SK（数学原理：两个大质数相乘，积很难因式分解） |
| 15 | 4. 速度慢，只对少量数据加密                            |

对称性加密：加密解密都是使用相同的密钥

AES加密，DES加密

（拿着公钥是可以进行加密也可以进行解密）

非对称性加密：加密和解密是使用不同的密钥

拿着公钥只可以进行加密无法进行解密，只有拿着私钥可以进行解密，

拿着私钥只可以进行解密无法进行加密，只有拿着公钥可以进行加密

（私钥可以推测出公钥，但是公钥无法推测出私钥——是不可逆的）

## 十六、简单说一下anr和crash产生的原因有哪些？

### App出现ANR，是什么原因导致的？

（Application Not Responding应用无响应）

- |   |                      |
|---|----------------------|
| 1 | 1.耗时的网络访问            |
| 2 | 2.大量的数据读写            |
| 3 | 3.服务器繁忙导致连接超时无响应     |
| 4 | 4.其他线程持有锁，导致主线程等待超时  |
| 5 | 5.其他线程终止或崩溃导致主线程一直等待 |

### App出现crash原因有哪些？

- |   |        |
|---|--------|
| 1 | 内存过低   |
| 2 | 内存泄漏   |
| 3 | 设备不兼容  |
| 4 | 程序逻辑错误 |
| 5 | 网络因素   |

1、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；

2、是内存泄漏，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。亦或非授权的内存位置的使用也会可能导致app crash；

3、程序逻辑错误：数组越界，堆栈溢出，并发操作，逻辑错误；

5、设备兼容：由于设备多样性，APP在不同设备上可能会有不同的表现



4、网络因素：可能网速欠佳，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会影响APP的稳定性

APP对于不稳定偶然出现anr和crash的时候你是怎么处理的

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

## UI自动化相关知识

---

### 一、八种定位方式都有哪些？

#### 1.Id 定位

`driver.find_element_by_id("zentao").click()` click是点击

`driver.find_element_by_id("account").send_keys("admin")`

通过id定位，然后点击进行运行测试，并输入内容

-----

#### 2.Name定位

`driver.find_element_by_name("account").send_keys("admin")`

通过name定位，然后点击进行运行测试，并输入内容

`driver.find_element_by_name("username").send_keys("test")`

输入用户名 "username"是唯一值，然后继续查询，再点击，输入账户号

`driver.find_element_by_name("password").send_keys("123456")`

输入密码 "password"是唯一值，然后继续查询，再点击，输入密码

-----

#### 3.全文本定位

`driver.find_element_by_link_text("开源版").click()`

通过元素来定位

-----

#### 4.部分文本定位

`driver.find_element_by_partial_link_text("开源").click()`

通过元素来定位

-----

## 5.Class name定位

```
driver.find_element_by_class_name("us_Submit").click()
```

点击登录 通过class属性进行去定位元素，再进行查询，最后点击进入

---

## 6.通过标签名来定位

```
driver.find_element_by_tag_name("input").send_keys("admin")
```

---

## 7.Xpath定位

```
driver.find_element_by_xpath("/html/body/div/div/div[1]/div/div/div[2]/a[1]").click()
```

根据Xpath来定位查询测试 绝对路径定位查找

```
driver.find_element_by_xpath("//a[@id='zentao']").click()
```

相对定位查找

---

## 8..Css定位

```
driver.find_element_by_css_selector("#zentao").click()
```

根据CSS来定位查询 css里面有id的话 #id的值 有class .class的值 用其他属性 [属性="属性值"]

```
driver.find_element_by_css_selector(".us_Submit")
```

## 二、手工测试与自动化测试的优缺点？

测试工作不论是手工测试还是自动化测试都是软件质量保证的一个途径，对于一些基本的逻辑性不强的操作，可以使用自动化测试工具，对于一些逻辑性很强的操作，如果自动化测试不是很健全，不建议使用，这时就需要手工测试了，手工测试也存在着一些缺陷，手工测试最常做的是重复的手工回归测试，不但代价昂贵而且容易出错

### 三、自己会不会搭建测试环境？

在工作中我搭建过自动化测试环境，我去公司之前已经搭建好了，我们公司有一个十年工作经验的自动化测试工程师，他搭建的自动化测试框架，公司已经有完整的自动化测试环境搭建手册，我们只需要按照手册一步一步安装需要的软件，以及配置相应环境变量等等，比如，我们公司采用的是Python + selenium + unittest + ddt框架以及POM设计模式(PO：page object model页面对象模型)，来进行自动化测试的，我们先安装python，安装selenium，接下安装pip安装pillow，ddt，pymysql，HTMLTestRunner等第三方库，安装Pycharm，然后配置相应环境变量就可以了。

### 四、POM的优势是什么？

POM提供在UI层操作，业务流程与验证分离的模式，这使得测试代码变得清晰与高可读性，对象库与用例分离，使得我们更好的复用对象，甚至能与不同的工具进行深度的结合应用，可复用的页面方法代码会变得更加优化。

### 五、功能测试与自动化测试个占多少比例？

一般自动化测试不会超过30%，还是以功能测试为主，自动化和手工测试的比例大概是3：7左右

### 六、在自动化测试框架中怎么组织测试用力的？

(怎么识别脚本中的case的？)

我们的测试用例都是放在case的目录下，通过unittest单元测试框架里的discover的方法筛选出以test开头的用例，循环添加到测试套件中让系统全部执行筛选出的用例。

### 七、你们的测试数据是怎么进入到测试用例的？

我们首先通过xlrd第三方库来封装读取excel文件里的方法，通过封装的方法读取放在data目录下的excel数据文件，放在一个列表里，通过ddt数据驱动框架，将列表中的数据循环添加到测试用例里

### 八、你们的测试用例维护在哪里？

删除过时的测试用例

改进不受控制的测试用例

删除冗余的测试用例

增添新的测试用例

## 九、自动化测试中下拉表示如何处理的？

首先我们处理这类问题，我们可以把他的只读属性去掉，然后就可以正常输入了，比如，先引入select，然后定位下拉框这个元素的位置,然后把定位到的元素引入到这个

select,然后根据下拉列表里面的内容来选择输入(比如列表里是文本内容的选择)

然后通过select这个方法选择列表里面的文本内容就可以了。

## 十、自动化如何进行截图？

调用get\_screenshot\_as\_file的方法进行截图，括号里的是保存到哪个盘且命名

#页面全部截图

```
driver.get_screenshot_as_file("D:\geeknet.png")
```

## 十一、如何截取部分图片？

首先我们要引入PIL方法里的image

- 1、打开界面后进行截图保存
- 2、获取想要的截图位置以及获取元素的位置
- 3、使用location方法获取位置、左上角坐标，图片的大小以及宽高
- 4、横坐标加宽度为右下角的横坐标，纵坐标加高度为右下角纵坐标
- 5、调用image里的open打开截图
- 6、在crop按照坐标截图
- 7、Save保存截图

## 十二、python如何连接数据库？

**十三、如何处理时间控件？**

**十四、如何处理弹出框的？**

**十五、如何处理windows自带的上传框？**

**十六、自动化测试怎么导出html测试报告？**

**十七、UI自动化测试框架是什么？**

**十八、自动化测试跑一个流程需要多久？**

**十九、自动化测试主要实现哪些功能？**

**二十、如何提高自动化测试脚本的稳定性？**

( 用例在运行过程中经常出现不稳定的情况也就是说这次通过，下次没有办法通过，如何提升用例的稳定性？如何提高selenium脚本的执行速度，不管网络加载慢还是快，selenium中如何保持操作元素的成功率？ )

**二十一、什么样的项目适合做自动化测试？**

**二十二、对于验证码是如何处理的？**

**二十三、元素等待都与哪些？**