

# 软件测试面试宝典

## 目 录

|                                     |    |
|-------------------------------------|----|
| 一、引言 .....                          | 9  |
| 1.1 文档目的 .....                      | 9  |
| 二、职业规划 .....                        | 9  |
| 2.1 简单的自我介绍下 .....                  | 9  |
| 2.2 为什么离职 .....                     | 9  |
| 2.3 加班的看法 .....                     | 11 |
| 2.4 你还有什么问题要问? .....                | 11 |
| 2.5 你的职业发展规划和职业目标 .....             | 11 |
| 2.6 你最近测试的论坛和网站是什么? .....           | 12 |
| 2.7 对于一个新的工具你有什么看法 .....            | 12 |
| 2.8 你做了 3 年测试收获，你的测试心得 .....        | 12 |
| 2.9 怎么转行的，毕业之后一直从事软件测试吗 .....       | 12 |
| 2.10 以前在哪里上班，公司地址，测试多少人，开发多少人 ..... | 12 |
| 2.11 你的期望薪资是多少? .....               | 12 |
| 2.12 多久能入职? .....                   | 13 |
| 2.13 是怎么考虑做软件测试呢? .....             | 13 |
| 2.14 是否有考虑往开发方面测试? .....            | 13 |
| 2.15 计算机专业课程有哪些? .....              | 13 |
| 2.16 说一下上一个公司背景? .....              | 13 |
| 2.17 超过 26 岁女孩子，目前有没有打算结婚 .....     | 13 |
| 三、测试理论 .....                        | 13 |
| 3.1 你们原来项目的测试流程是怎么样的? .....         | 13 |
| 1、需求了解分析阶段 .....                    | 14 |
| 2、测试准备阶段 .....                      | 14 |
| 3、测试执行阶段 .....                      | 14 |

|   |    |
|---|----|
| 3.2 如果需求不明确的话你怎么办?.....                   | 14 |
| 3.3 有哪些需要评审, 哪些人在 .....                   | 14 |
| 3.4 有没有写过测试计划, 具体包括哪些内容?.....             | 14 |
| 3.5 用例包含哪些部分, 哪些用例设计方法, 你一般常用哪些方法?.....   | 15 |
| 3.6 TestLink 工具使用?.....                   | 16 |
| 3.7 如何提交一个好的 BUG.....                     | 16 |
| 3.8 提 bug 需要注意哪些问题?.....                  | 16 |
| 3.9 bug 怎么管理的, bug 的生命周期或者是 bug 的状态 ..... | 16 |
| 3.10 提交 bug 包含哪些内容 .....                  | 16 |
| 3.11 你提交的 bug, 开发不认可怎么办?.....             | 17 |
| 3.12 对应无法重现 bug, 应该怎么处理?.....             | 17 |
| 3.13 界面中的乱码可以是哪里导致的?.....                 | 17 |
| 3.14 bug 的级别有哪些, 级别如何判断 .....             | 17 |
| 3.15 测试中, 如何判断是前端的 bug 还是后端的 bug 呢?.....  | 17 |
| 3.16 项目上线后发现 bug, 测试人员应该怎么办 .....         | 17 |
| 3.17 如何保证质量 .....                         | 18 |
| 3.18 产品是怎么上线的?.....                       | 18 |
| 四、项目 .....                                | 18 |
| 4.1 简单介绍下最近做过的项目 .....                    | 18 |
| 4.2 拿一个你所负责的模块, 讲下具体怎么测的?.....            | 19 |
| 4.3 你在这个项目里面主要做了些什么工作? .....              | 19 |
| 4.4 你们项目组有多少人、开发多少个、测试多少个?.....           | 19 |
| 4.5 测试人员怎么分工的?.....                       | 19 |
| 4.6 项目的迭代周期? 多久一迭代? 一个版本你们发现多少 bug.....   | 19 |
| 4.7 你们整个项目写了多少用例, 你负责的模块大概写了多少用例?.....    | 20 |
| 4.8 最近的版本写了多少用例?.....                     | 20 |
| 4.9 你的需求分析一般几天, 用例大概写了多长时间?执行了多长时间?.....  | 20 |
| 4.10 在 uat 测试的时候, 突然客户临时要大量的数据 .....      | 20 |
| 4.11 发现哪些映像比较深刻 bug/经典 bug?.....          | 21 |
| 4.12 每个阶段测试开发在干嘛 (比如你写用例的时候开发在干嘛? ) ..... | 23 |
| 4.13 你们公司是否敏捷开发 .....                     | 23 |

|  |    |
|--|----|
| 4.14 你这个项目做了多久？你这个项目现在的用户里有多少？活跃量多少？.....      | 24 |
| 五、测试思维 .....                                   | 24 |
| 5.1 打电话功能怎么去测？ .....                           | 24 |
| 5.2 给你一个杯子怎么测？ .....                           | 25 |
| 5.3 图像上传功能的测试点 .....                           | 26 |
| 5.4 搜索框的测试 .....                               | 27 |
| 六、Linux .....                                  | 28 |
| 6.1 Linux 系统你是怎么用的？ .....                      | 28 |
| 6.2 Linux 是什么工具连接的，Linux 版本是多少 .....           | 29 |
| 6.3 日志具体是怎么看的 .....                            | 29 |
| 6.4 监控资源命令用到哪些 .....                           | 29 |
| 6.5 nmon 工具是如何使用的 .....                        | 29 |
| 七、数据库 .....                                    | 30 |
| 7.1 你们数据库怎么用 .....                             | 30 |
| 7.2 存储过程是怎么编写的 .....                           | 30 |
| 7.3 常见的关系型数据库有哪些 .....                         | 31 |
| 7.4 你们用的什么数据库连接工具 .....                        | 31 |
| 7.5 左连接与右连接有什么区别 .....                         | 31 |
| 7.6 索引有哪些，如何建立索引，索引的优缺点 .....                  | 31 |
| 八、抓包与网络协议 .....                                | 31 |
| 8.1 抓包工具怎么用 .....                              | 31 |
| 8.2 如何抓取 https 的包 .....                        | 32 |
| 8.3 如何抓取手机的包 .....                             | 32 |
| 8.4 网络协议了解多少？ .....                            | 32 |
| 8.5 请求方式有哪些？ .....                             | 33 |
| 8.6 get 跟 post 请求的区别 .....                     | 33 |
| 8.7 http 跟 https 的区别 .....                     | 33 |
| 8.8 为什么要使用 cookie 和 session: http 是无状态协议 ..... | 33 |
| 8.9 cookie 跟 session 的区别 .....                 | 33 |
| 8.10 post 申请方式，用 get 会报什么错误。 .....             | 34 |
| 8.11 http 协议提交请求头内容 .....                      | 34 |

|  |    |
|--|----|
| 九、接口测试.....  | 34 |
| 9.1 接口测试怎么测 .....                                      | 34 |
| 9.2 两个接口有关联， jmeter 具体怎么做 .....                        | 36 |
| 9.3 接口测试主要目的是什么? .....                                 | 36 |
| 9.4 接口测试的流程 .....                                      | 36 |
| 9.5 接口测试和平常的 UI 测试有什么区别? .....                         | 37 |
| 9.6 给你一个新的接口，你怎么去设计用例? .....                           | 37 |
| 9.7 接口文档主要包含哪些内容? .....                                | 37 |
| 9.8 你们什么时候测试接口 .....                                   | 38 |
| 9.9 你怎么去检查，分析 .....                                    | 38 |
| 9.10 什么是 api 接口测试 .....                                | 38 |
| 9.11 什么情况下开展接口测试? .....                                | 38 |
| 9.12 依赖于第三方的接口如何测试 .....                               | 38 |
| 9.13 你们接口怎么鉴权的? .....                                  | 39 |
| 9.14 接口传输格式有哪些 .....                                   | 39 |
| 9.15 cookie、session、token 的区别 .....                    | 40 |
| 9.16 接口测试的工具具有哪些? .....                                | 40 |
| 十、接口自动化 .....  | 40 |
| 10.1 接口自动化怎么测试 .....                                   | 40 |
| 10.2 为什么做接口自动化? .....                                  | 42 |
| 10.3 假如公司想要做自动化，让你去做，你会从哪些方面考虑入手? .....                | 42 |
| 10.4 你写了多少接口自动化用例 .....                                | 42 |
| 10.5 比如说你接口的请求参数需要加密处理的，你们用的是什么加密方式，你加密怎么处理的?<br>..... | 43 |
| 10.6 你查询出来返回结果是密文，密文你怎么测试 .....                        | 43 |
| 10.7 http 如何进行代理录制接口 .....                             | 43 |
| 10.8 jmeter 如何进行参数化，参数化类型包含哪些 .....                    | 43 |
| 10.9 jmeter 中对于 json 数据如何提取信息 .....                    | 43 |
| 10.10 jmeter 中如何跨线程组传输参数 .....                         | 43 |
| 10.11 jmeter 如何进行断言 .....                              | 44 |
| 10.12 jmeter 如间在 cmd 命令下运行 .....                       | 44 |

|       |   |    |
|-------|---|----|
| 10.13 | jmeter 运行完成后如何去自动发送邮件?                        | 45 |
| 10.14 | pytest 如何做断言?                                 | 45 |
| 10.15 | patent 中如何去调取其他用例中返回的参数?                      | 45 |
| 10.16 | 你们做接口自动化, 用例数据是怎么组织, 管理的?                     | 46 |
| 10.17 | requests 中如何进行动态关联                            | 46 |
| 10.18 | 你们 python 接口自动化怎么去处理 cookie, session 的?       | 46 |
| 十一、   | UI 自动化  | 46 |
| 11.1  | ui 自动化怎么测试                                    | 46 |
| 11.2  | 什么是自动化测试?                                     | 49 |
| 11.3  | 什么样的项目合适做自动化(至少三个)                            | 49 |
| 11.4  | 自动化测试流程是什么?                                   | 49 |
| 11.5  | 自动化实现的比例, 1-2 年。                              | 49 |
| 11.6  | 自动化测试用例的来源                                    | 49 |
| 11.7  | 自动化测试的优点与缺点                                   | 49 |
|       | 优点:   | 49 |
|       | 缺点:   | 50 |
| 11.8  | 你们自动化覆盖率有多少?                                  | 50 |
| 11.9  | 自动化测试环境的搭建是怎样的?                               | 50 |
| 11.10 | seleniun 的原理是什么?                              | 50 |
| 11.11 | Selenium2 与 Selenium1 的区别是什么?                 | 51 |
| 11.13 | 定位元素的 8 个方法是什么?                               | 51 |
| 11.28 | 定位不到元素, 你碰到过哪些?怎么解决的?                         | 52 |
| 11.29 | 元素定位, 有时候定位得到, 有时候定位不到, 可能是什么原因, 你会怎么处理?..... | 52 |
| 11.30 | 对于不可见/隐藏的元素, 你如何定位, 如何处理?                     | 52 |
| 11.31 | 如果元素经常变动你怎么做?                                 | 52 |
| 11.32 | 自动化如何破解验证码                                    | 53 |
| 11.33 | 数据驱动有没有了解过, 具体怎么做的?                           | 53 |
| 11.34 | 参数化的类型有哪些?自动化脚本如何实现数据的参数化?                    | 53 |
| 11.35 | 随机数如何操作?                                      | 53 |
| 11.36 | 自动化中如何去操作 excel 表格?                           | 53 |
| 11.37 | 如何去读取表格中一行, 一列数据, 具体数据                        | 54 |

|       |  |    |
|-------|--|----|
| 11.38 | 如何读取表格中总的列数跟行数 .....                               | 54 |
| 11.39 | 自动化中如何去操作数据库 .....                                 | 54 |
| 11.40 | 如何获取数据库返回的所有，一行，前 5 行 .....                        | 54 |
| 11.41 | 数据库返回的数据格式是什么类型的格式 .....                           | 55 |
| 11.42 | ui 自动化你是怎么做断言的? .....                              | 55 |
| 11.43 | 断言有哪几种(三种)，常见 assert 断言 .....                      | 56 |
| 十二、   | Unittest 框架 .....                                  | 56 |
| 12.1  | 你们自动化用例是怎么管理的? .....                               | 56 |
| 12.2  | Web UI 自动化都用到过哪些库? .....                           | 56 |
| 12.3  | Unittest 框架的原理? .....                              | 56 |
| 12.4  | Unittest 框架有哪些组件? .....                            | 57 |
| 12.5  | Unittest 框架如何使用? .....                             | 57 |
| 12.6  | 如何生成自动化测试报告? .....                                 | 58 |
| 十四、   | 性能测试 .....   | 59 |
| 14.1  | 性能测试怎么测试 .....                                     | 59 |
| 14.2  | 性能测试流程是怎么样的? .....                                 | 60 |
| 14.3  | 你们性能观察哪些指标，大概指标范围是怎么样的。 .....                      | 61 |
| 14.4  | 这个测试的环境配置，如转速度 .....                               | 61 |
| 14.5  | 性能测试计划有哪些内容 .....                                  | 61 |
| 14.6  | 有没有写过性能测试报告，具体包括哪些内容 .....                         | 62 |
| 14.7  | 什么是内存泄漏，什么是内存溢出? .....                             | 62 |
| 14.8  | 吞吐量，吞吐率 .....                                      | 62 |
| 14.9  | 吞吐量与吞吐率跟负载有什么关系? .....                             | 63 |
| 14.10 | 当你服务器满了之后，你们吞吐量和响应时间怎么变化的 .....                    | 63 |
| 14.11 | 你们的 TPS 的指标是什么估算的? .....                           | 63 |
| 14.13 | 每人说一个项目接口，你设置多少并发 .....                            | 65 |
| 14.14 | 你们吞吐量是多少，响应时间是多少，你设置了多少并发? .....                   | 66 |
| 14.15 | 做并发你们一般 cpu 和内存是多少? .....                          | 66 |
| 14.16 | 有没有做过稳定性测试 .....                                   | 66 |
| 14.17 | 5000 个人抢购，只能 50 个人能抢到，你怎么设计并发数的 .....              | 67 |
| 14.18 | 微信群里面发送红包，5000 个人群，只能 3000 个人能抢到，你怎么设计并发数的峰值 ..... | 67 |

|                 |   |    |
|-----------------|---|----|
| 14.19           | 20 并发 40 次循环怎么做？ .....                  | 67 |
| 14.20           | 我想从 200 慢慢加载到 300，到 400 怎么做 .....       | 67 |
| 14.21           | 需要插入 500 条数据，你怎么插入 .....                | 67 |
| 14.22           | 响应超时，你是怎么定位的 .....                      | 67 |
| 14.23           | 压测返回数据报错，你怎么去定位的 .....                  | 67 |
| 14.24           | 你理解的性能调优是什么？ .....                      | 68 |
| 14.25           | 如果要做万并发，你怎么做 .....                      | 69 |
| 14.26           | 如果用户并发要慢慢加载，你怎么设置的 .....                | 69 |
| 14.27           | 并发用户数跟响应时间与吞吐的关系 .....                  | 69 |
| 十五、app 测试 ..... |   | 69 |
| 15.1            | app 测试你具体怎么做的？ .....                    | 69 |
| 15.2            | Web 测试与 app 测试区别？ .....                 | 70 |
| 15.3            | 常用的 adb 的命令？ .....                      | 70 |
| 15.4            | adb 的作用的？ .....                         | 71 |
| 15.5            | App 兼容性测试怎么做的？ .....                    | 72 |
| 15.6            | App 稳定怎么做的？ Monkey 怎么用(App 稳定测试)？ ..... | 72 |
| 15.7            | App 弱网测试怎么做的 ？ .....                    | 73 |
| 15.8            | App 的性能测试 .....                         | 74 |
| 15.9            | 对于内存具体怎么测呢？ .....                       | 75 |
| 15.10           | 对于 CPU 具体怎么测呢？ .....                    | 75 |
| 15.11           | 对于流量具体怎么测？ .....                        | 75 |
| 15.12           | 对于 FPS 具体怎么测？ .....                     | 75 |
| 15.15           | App 交互性怎么做的？ .....                      | 76 |
| 15.16           | App 的安装，卸载，更新测试具体从哪些方面考虑？ .....         | 76 |
| 15.17           | H5 界面怎么测试 .....                         | 77 |
| 15.18           | 你们之前是用什么手机什么版本做兼容性测试的？ .....            | 79 |
| 15.19           | Android 跟 ios 测试有什么区别？ .....            | 79 |
| 15.20           | 小程序怎么测试 .....                           | 79 |
| 15.21           | 公众号，小程序比 app 更火，你怎么看 .....              | 80 |
| 15.22           | 微信开发者工具如何使用 .....                       | 80 |
| 十六、安全测试 .....   |   | 80 |

|   |    |
|---|----|
| 16.1 安全测试怎么测试?                          | 80 |
| 16.2 appscan 的原理是什么                     | 82 |
| 十七、python 编程                            | 82 |
| 17.1 一个变量没有定义报什么错误                      | 82 |
| 17.2 列表与元组的区别                           | 82 |
| 17.3 python 序列有哪些?                      | 83 |
| 17.4 Python 区分大小写吗?                     | 83 |
| 17.5 如何获取字典中的所有键?                       | 83 |
| 17.6 列表和字典有什么区别?                        | 83 |
| 17.7 截取字符串里面部分字符怎么做?                    | 83 |
| 17.8 如何遍历列表, 字典?                        | 84 |
| 17.9 列出 5 个 python 标准库?                 | 84 |
| 17.10 字典如何删除键和合并两个字典?                   | 84 |
| 17.11 python 实现列表去重的方法?                 | 85 |
| 17.12 python 内建数据类型有哪些?                 | 85 |
| 17.13 列出 python 中可变数据类型和不可变数据类型, 并简述原理? | 85 |
| 17.14 如何将字符串转换为小写?                      | 86 |
| 17.15 回文数                               | 86 |
| 二十、场景问题                                 | 86 |
| 20.1 有没有抓过包, 为什么要抓包, 具体怎么做的?            | 86 |
| 20.2 为什么要进行抓包?具体怎么分析的?                  | 87 |
| 20.3 大概给我讲下抓包的原理?                       | 87 |
| 20.4 给你一个需求, 你具体讲下, 你怎么测试?              | 87 |
| 20.5 给你一个项目, 你怎么展开工作, 请详细说一下, 你具体怎么做的   | 87 |
| 20.6 项目快上线因为开发延迟, 测试时间被压缩怎么做?           | 88 |
| 二十一、测试经理管理                              | 88 |
| 21.1 员工的绩效考核, 量化标准                      | 88 |
| 21.2 你组内有一个技术比你牛逼年轻的组员不服你, 你会怎么处理       | 89 |
| 21.3 你的组员提交的 bug 开发不认同而吵起来了, 你怎么处理      | 89 |



## 一、引言

### 1.1 文档目的

本次文档是为了方便<软件测试>课程学生，在面试中遇到的一问题，与常见的一些答案并不是唯一答案

## 二、职业规划

### 2.1 简单的自我介绍下

面试官，您好，我叫 XXX，来自于 XXXX，目前从事软件测试工作，已经三年工作经验，个人性格，比较开朗，跟人关系比较好，做事也比较细心三年测试工作经验中，过了不少项目，积累不少项目经验，前面 1-2 年主要是功能测试，后面这一年主要做接口测试，app 自动化测试能够独立完成软件产品测试工作，能够独立编写测试文档，包括用例，计划，报告等，熟悉 linux 跟数据库，熟悉 jmeter 与 python + request 进行接口测试，也可以使用 pytest 框架进行接口自动化测试，python + selenium + pytest 框架进行自动化测试，python + appnium + pytest 移动 app 自动化测试框架，熟悉使用 Jenkins 持续集成，熟悉 app 专项测试与小程序测试，熟悉抓包工具。

我个人平常喜欢…看书…

我个人觉得测试这一块，主要是对需求了解，需求理解到位，工作当中，一定细心耐心，技术这块，不断学习能力

如果面试没有说话，这个是我的一个简单自我介绍，看面试官还有什么需要了解的

### 2.2 为什么离职

①、尽量淡化敏感答案比如，人际关系不好处理，与上司相处不好、工作压力大等。

人际关系复杂，现代企业讲求团队精神，要求所有成员都能有与别人合作的能力，你对人际关系的胆怯和避讳，可能会被认为是心理状况不佳，处于忧郁焦躁孤独的心境之中，从而妨碍了你的从业取向。

收入太低，这样回答会使对方认为你是单纯为了收入取向，很计较个人得失，并且会把“如有更高的收入，会毫不犹豫地跳槽而去的”这种观念形成对你的思维定势。

分配不公平。现在企业中实行效益薪金、浮动工资制度是很普遍的，旨在用物质刺激手段提高业绩和效率；同时，很多单位都开始了员工收入保密的措施，如果你在面试时将此作为离开原单位的借口，则一方面你将失去竞争优势，另一方面你会有爱打探别人收入乃至隐私的嫌疑。

工作压力太大。现代企业生存状况是快节奏的，企业中的各色人等皆处于高强度的工作生存状态下，有的单位在招聘启事上干脆直言相告，要求应聘者能在压力下完成工作，这是越来越明显的趋向

②、尽量采用与工作能力关系不大、能为人所理解的离职原因。

寻求更大的发展：现有的企业岗位设置难以满足自身职业进一步发展的要求，

换一个更好的平台来挑战自我。

原公司发生了重大客观变化：公司重组或部内部变动，导致工作内容发生重大变化，

无法再继续履行劳动合同，或者直接被裁员等

与公司文化无法融合：每一家企业其实都有自己特有的“文化”，如果你在这家公司里工作，

却无法认可这家公司提倡的一些文化，这就会对企业的发展以及对自己的发展都非常的不利。

所以你想要走出企业的束缚，找到一家跟你更契合的公司工作也是可以的。

个人原因：上班太远影响工作、家中有事、充电、休假、生病等等。注意：避免敏感答案，

并不意味着欺骗，如招聘人员问及细节问题，应如实回答。否则求职者的诚信度可能大打折扣，

成功可能性更小。

### ③、不要诋毁你的老东家

相信很少有人会犯这样的错误，这的确是一个不可取的方式，你应该把你的离职原因集中表述

在“寻找新机会或新的平台”以及尝试在新的岗位上提升自己。职场虽然没有战争那么血腥，

但也有很多委屈、不被理解或被无故受伤，这些都很正常的。我们要用一颗阳光的心去面对，

用一颗阳光的心去照耀。

在离职后，永远要用赞美的词语来评价你的老东家，哪怕他在公开场合去骂你，你都要用最美的词语

去评价他。别忘记，你赞美他，是你素养高；他骂你，是他素质的问题。你永远改变不了别人，但你

有能力改变自己。

### ④、体现你的忠诚度

如果你轻描淡写地就离开了之前的团队，那么面试官会觉得你在新公司也可能会轻易走人，

所以，在体现忠诚度的时候，你可以试着谈谈你离开上一家公司时有多么痛苦依依不舍（即使并没有），

聊聊如果有办法使你能在原来的岗位上持续得到提升或者如果不是因为股东之间的权利斗争（可适当

显得痛心疾首些），你肯定不会离开。

而在体现责任感时，你需要表达两层意思：

首先，你从上家公司离职时已经为继任者做了充分的交接。你需要清楚地表明：你在上家公司也很认

真尽职，并且同事之间一直保持互助互利的工作氛围。也许你可以说说你也想过要早些辞职，但是考

虑某个未完成的重要项目、或是继任者短期内还不能胜任角色所以晚了一些。

其次，就是你很期望承担新的职责，并表现出你的热忱，这种热忱除了对工作的热忱之外，

也可以适当的通过向你的面试官（不仅是 HR）提问表现出来对面试官的兴趣、对他们技能的

认可以及共鸣，例如，“那么，你是如何做到现任职位的？”或：“如果我有幸担任这个职位，

你会给我哪些建议？”

通用说法：家人在这边，或者想到大城市发展。

## 2.3 加班的看法

1、(将问题抛给 HR，)在回答您的问题之前我想了解一下贵公司的加班制度是怎样的。

(这种回答，其实是把问题抛回给 HR，让 HR 表明公司对于加班的态度，其实很多大公司对于加班都有明确的规定，什么情况该加班，加班会有什么福利等等都是确定的，而小公司就随意很多，往往是老板要求加班员工就得无条件加班。因此，如果公司的加班制度明确，那么 HR 就能够明确地向你介绍，你在了解过后再给出回应也不迟。面试本就是双向选择的过程，你也没必要为了通过面试，满口答应自己愿意加班。)

2、在 IT 行业里，加班是比较正常的，首先我会了解加班的原因，如果公司近段时间需要赶项目进度或者是站在重要的关键节点上需要加班，我会站守自己的岗位，把自己手上的事做好，和团队一起加班，让公司按预期的进度推进项目，这在我看来是必要的加班，如果是其他的原因，比如个人原因，我会努力不加班，在保证工作质量的前提下，我会提高自己的工作效率，避免加班。

如果说公司基本上天天都要加班，加班的频车较高，我希望可以减少不必要的加班，让员工得到充分的休息，有休息才会把工作效率提升上来，工作才会更有效率，另外有些工作上的能力可以在其它地方，而不在工作的本身，比如对生活的理解和感悟等之类，是从工作中学习不到的，正所谓功夫在诗外嘛。

还有，我之前在上一个公司上班，住的地方离公司较远，作为一个女生，出于个人人身安全考虑，我更希望不加班，毕竟生命健康是从事一切活动的前提嘛。

通用说法：如果项目组比较忙，加班是没有问题的。

## 2.4 你还有什么问题要问？

少问一些福利相关的问题

1. 公司现在做什么项目
2. 公司目前做哪方面测试
3. 公司这边测试人员分配比例
4. 进入公司，我这边大概的工作安排
5. 公司这么后续发展机会还有培养
6. 有没有培训
7. 面试没有回答上的问题，再去请教

## 2.5 你的职业发展规划和职业目标

根据公司况，个人原因来说：看公司的岗位要求，招岗位就是攻能  
公司只做功能测过

接下来一年时间内，更加完善自己的功能测试，2-3 年内熟自动化或者性能，  
3-5 年内系统成为自动化或者性能，成为资深技术人员

公司做自动化与性能测试

1-2 年内熟悉自动化或者性能，3-5 系统成为自动化或者性能，成为资深技术人员性能与自动测试  
找一个比较稳定平台，跟公司长期发展，后期走管理或者产品路线

## 2.6 你最近测试的论坛和网站是什么？

工作中积累，查看网站论坛(51 testing)，CSDN，书籍《性能测试专家》，《性能之巅》  
偏开发，python 自动化，selenium 自动化

## 2.7 对于一个新的工具你有什么看法

1. 自己先去研究 2. 找会的去请教 3. 百度去找资料 4. 工具原始文档

## 2.8 你做了 3 年测试收获，你的测试心得

3 年的测试经验，对我来说也是 3 年的工作经验，在这 3 年的工作经验当中，我觉得态度比能力要重要，做好一个测试，最主要是性格，信心，耐心，细心还需要良好的沟通能力。不断学习的能力，产品质量，测试流程这块很关键，好的计划加好的执行才能成就好的产品。

## 2.9 怎么转行的，毕业之后一直从事软件测试吗

非专业

- 1、培训，不要说刚培训出来 2019 年底开始
- 2、自学，不断一直学习
- 3、家里有关系，带你入行，后面学习过程很勤奋

计算机相关专业

- 1，实习开始，公司分配到做测试，做测试过程，比较喜欢测试，一直做下来

## 2.10 以前在哪里上班，公司地址，测试多少人，开发多少人

根据你简历上面写的，公司详细地址

产品 1、项目 1 个、架构师 1 个、前端 3 个、后端 5 个、Oos 1 个、Android 1 个，  
测试 3 个(测试主管，核心测试人员)、运维 1 个、ui 1 个

## 2.11 你的期望薪资是多少？

- 1，期望薪资不要说区间比如说：6-8，那肯定是 6
- 2，如果原来公司在二线城市原有薪资不要说太高，
- 3，深圳那边薪资，比广州高 10% - 15%

## 2.12 多久能入职?

- 1、公司比较满意，直接随时过去。
- 2、不是很满意，下周一，个人有点事情，比如说回家一趟等等。

## 2.13 是怎么考虑做软件测试呢?

- 1、个人性格合适
- 2、前景还可以
- 3、个人技能也匹配

## 2.14 是否有考虑往开发方面测试?

我觉得，IT 行业，没有具体的界限，后期，开发也要懂测试，测试也要懂开发，如果公司有机会，愿意去尝试

## 2.15 计算机专业课程有哪些?

公共课程：数学(高等数学、线性代数、概率论与数理统计、离散数学、数值分析)，  
政治(马克思主义思想概论、毛泽东思想概论与中国特色社会主义思想、  
思想道德修养与法律基础、中国近现代史纲要)、大学英语、体育。

专业基础课程：电路原理、模拟电子技术、数字逻辑、微机原理、汇编语言、操作系统原理、  
编译原理、算法与数据结构、面向对象方法、C 语言/C++语言等。

专业方向课程：计算机数据库原理、python 语言、图形学、人工智能，多媒体技术、  
网络安全、人机交互、无线互联网技术、软件开发方法、高性能技术系统仿真和虚拟现实等。  
大专 3 年、本科 4 年，  
本科四级、大学英语四级及格 425 分、总分 710

## 2.16 说一下上一个公司背景?

了解公司主要是什么项目，百度查下，  
如果公司主营产品跟你项目不匹配  
比如：原来公司做医疗设备，那就说：我们是项目外包的部门，专门接项目

## 2.17 超过 26 岁女孩子，目前有没有打算结婚

暂时没有结婚的计划与打算，如果已经有小孩，说暂时不考虑二胎，  
有小孩，可以说小孩在老家(原来有学员因为这个问题，被公司 pas 过)

## 三、测试理论

### 3.1 你们原来项目的测试流程是怎么样的?

我们的测试流程主要有三个阶段：需求了解分析、测试准备、测试执行。

## 1、需求了解分析阶段

我们的 SE 会把需求文档给我们自己先去了解一到两天这样，之后我们会会有一个需求澄清会议，我们会把不明白不理解的需求在会议上说出来，包含需求的合理性还有需求的可测性等，产品这边解答，目的是让我们测试这边和开发对需求的理解达到一致。

## 2、测试准备阶段

会议结束之后我们开始准备测试工作，我们测试这边会写一个测试计划，分配每个人负责的模块，然后我们就根据自己负责的模块用 xmind(思维导图)进行测试需求分析，分析测试点，以及编写测试用例，之后我们会在自己的组内先进行评审，评审修改之后还会在我们的项目组评审，评审完后进行修改测试用例。

## 3、测试执行阶段

开发人员编写好代码之后，我们会把代码包通过 Jenkins 部署到测试环境提测，进行 SIT 测试，在正式测试之前我们会先做一个冒烟测试，冒烟测试通过之后我们才转测，在执行测试的过程中，我们如果发现 bug 就会用 tapd(或者禅道)记录并且提交 bug，也会进行 bug 复测，以及回归测试，每一轮测试结束之后我们都会写一个测试报告，一般情况下，测试 4-5 轮之后会达到上线要求，当达到上线的标准后，测试报告会认为测试通过，上线前我们会做预发布测试，预发布通过后，由项目组与产品决定时间上线，上线完成，一周左右我们会写一个项目总结测试报告，总结我们在上一个版本中遇到的问题以及今后有哪些地方需要改进，在产品迭代过程中，我们会跑自动化用例来进行回归测试。

### 3.2 如果需求不明确的话你怎么办？

需求不明确的话我会在需求澄清会议上面提出来，问清楚这个需求只有明确需求，才能更好的完成工作，后续工作中还是不清楚，可以找产品再去确认这个需求。

### 3.3 有哪些需要评审，哪些人在

- 1、xmind 思维导图评审，主要是测试人员
- 2、测试用例需要评审，测试人员，开发人员，产品人员
- 3、需求文档，项目组所有的人员，都会到场

### 3.4 有没有写过测试计划，具体包括哪些内容？

参考答案 1：

测试计划内容：

- (1) 目的和范围
- (2) 规程
- (3) 测试方案和方法
- (4) 测试的准入和准出
- (5) 测试计划(流程、时间安排、对应人员)
- (6) 测试的环境配置和人员安排
- (7) 交付件

## 参考答案 2

我们公司之前按照考核要求写过测试计划，不过后面老大觉得太耽误工作进度，后面一般都不再写测试计划，而是写版本计划，这个在版本计划，每个人的任务列出来，负责人列出来，自己根据自己的情况分配时间，然后汇总，大家一起开个小会评审就可以了。

### 3.5 用例包含哪些部分，哪些用例设计方法，你一般常用哪些方法？

原来我们用例包含

测试项目，用例编号、测试标题、优先级、预置条件、操作步骤、测试数据、预期结果  
黑盒测试用例设计方法：主要是等价类、边界值、错误推测法、判定表、因果图、正交表、流程分析法、状态迁移法、异常分析法。

常用的：等价类、边界值、判定表、流程分析法、错误推测法。

等价类是指某个输入域的子集合，在该子集合中，

各个输入数据对于揭露程序中的错误都是等效的，

并合理地假定：测试某等价类的代表值就等于对这一类其它值的测试，因此，可以把全部

输入数据合理划分为若干等价类，在每一个等价类中取一个数据作为测试的输入条件，

就可以用少量代表性的测试数据取得较好的测试结果，

等价类划分可有两种不同的情况有效等价类和无效等价类。

边界值的话就是对等价类划分方法的补充。测试工作经验告诉我，大量的错误往往是发生在输入或输出范围的边界上而不是发生在输入输出范围的内部，因此的话针对各种边界情况来设计测试用例，可以查出更多的错误，使用边界值分析方法设计测试用例的话，首先应该确定边界情况，通常输入和输出等价类的边界，就是应着重测试的边界情况应当选取正好等于，刚刚大于或刚刚小于边界的值作为测试数据，而不是选取等价类中的典型值或任意值作为测试数据。

对于错误推断法，这个是基于经验和直觉推测程序中所有可能存在的各种错误，

从而有针对性的去设计测试用例的方法的，主要就是列举出程序中所有可能有的错误和容易发生错误的特殊情况去根据这些情况来选择测试用例，例如，在单元测试时曾列出的许多在模块中常见的错误以前产品测试中曾经发现的错误等，这些就是经验的总结。还有，输入数据和输出数据为 0 的情况。输入表格为空格或输入表格只有一行。这些都是容易发生错误的情况，可选择这些情况下的例子作为测试用例。

前面介绍的等价类划分方法和边界值分析方法都是着重考虑输入条件但都没有考虑输入条件之间的联系，相互组合等等的情况。考虑输入条件之间的相互组合，可能会产生一些新的情况，

但是要检查输入条件的组合并不是一件容易的事情，即使把所有输入条件划分成等价类，

他们之间的组合情况也相当多，因此的话可以考虑采用一种适合于描述对于多种条件的组合，

相应产生多个动作的形式来考虑设计测试用例，这就需要用到因果图(逻辑模型)。

因果图方法最终生成的就是判定表它适合检查程序输入条件的各种组合情况。

### 3.6 TestLink 工具使用?

- (1)创建用户，并给新创建的用户指定权限。
- (2)创建测试用例，对测试用例进行增、删、改、查
- (3)把测试用例关联到对应的测试计划中。
- (4)把测试用例指派给对应的测试人员。
- (5)对应的测试人员，查看被指派的测试用例，并执行测试用例。

### 3.7 如何提交一个好的 BUG

对 BUG 有一个清晰明了的描述： 详细描述 BUG 重现的步骤  
对于产生 BUG 的环境进行描述： 提交 BUG 相关的图片和日志；  
定位好 BUG 的等级： 将预期结果与实际结果进行对比。

### 3.8 提 bug 需要注意哪些问题?

- 1) 不要急着提交，先跟开发说明 bug 的情况，定位分析下 bug。  
是前端问题还是后端问题再去提交 bug。
- 2) 简单明了的概括 bug 标题，清晰的描述 bug 重现步骤，分析 bug 和预期正确结果，附加 bug 的截图或者日志。描述 bug 的时候。
- 3) 在不能确认该情况是否为 bug 的时候，可以请教其他人。
- 4) 提交完 bug 以后，后面还要跟踪 bug 修复情况。

### 3.9 bug 怎么管理的，bug 的生命周期或者是 bug 的状态

原来 bug 是用禅道来管理的

原来我们公司 bug，提交 bug 直接给对应的开发人员，对应开发人员修复完成，交给测试复测，复测通过关闭 bug，不通过打回给对应开发。

提交-开发人员(已激活未确认)-开发进行确认，状态变成已激活，已确认，开发修复完成，标注状态是已修复，测试人员复测通过，已关闭，打回给对应开发，已经激活。

### 3.10 提交 bug 包含哪些内容

所属产品、所属模块、所属项目、影响版本、指派人员  
截止日期、严重程度、优先级、bug 类型、bug 环境  
Bug 标题、重现步骤、附件



### 3.11 你提交的 bug，开发不认可怎么办？

首先我会再看需求文档，是不是我的理解有误，如果是我对需求理解错的话我就去关闭 bug。

如果是 bug 再去让其他测试人员看看听下他们的意见，然后自己先再三去复测，并且保存好截图和日志，确定这是一个 bug 之后我就去跟开发说明白，并且给他看 bug 重现的截图以及日志，如果开发还是不认可的话我就跟产品或项目经理说明白情况。

### 3.12 对应无法重现 bug，应该怎么处理？

首先，我会多测几次，测了好多次都无法重现的话我就先把 bug 挂起，并且留意一下，看看往后的测试中，如果在后面的测试中重现 bug 就激活，如果经过几个版本都还没发现的话就关闭 bug。

### 3.13 界面中的乱码可以是哪里导致的？

(1)数据库中的编码设置      (2)前端页面编码      (3)后台代码也会编码

### 3.14 bug 的级别有哪些，级别如何判断

- 1、致命：对业务有至关重要的影响，业务系统完全丧失业务功能，无法再继续进行，或业务系统丢失了业务数据且无法恢复，影响公司运营的重要业务数据出错。
- 2、严重：对业务有严重的影响，业务系统已经丧失可部分的重要的业务功能，或业务系统丢失了业务数据且可以恢复，一般业务数据出错。
- 3、一般：对业务有较小的影响，业务系统丧失了较少的业务功能，  
例如：界面错误，打印或显示格式错误。
- 4、提示：对业务没有影响，不影响业务过程正常进行，  
例如：辅助说明描述不清楚，提示不明确的错误提示。

### 3.15 测试中，如何判断是前端的 bug 还是后端的 bug 呢？

通常可以利用抓包工具来进行分析。可以从三个方面进行分析：请求接口、传参数、响应。

- 1) 请求接口 un 是否正确如果请求的接口 ur 错误，为前端的 bug
- 2) 传参是否正确如果传参不正确，为前端的 bug
- 3) 请求接口 u 和传参都正确，查看响应是否正确如果响应内容不正确，为后端 bug
- 4) 也可以在浏览器控制台输入 js 代码调试进行分析

### 3.16 项目上线后发现 bug，测试人员应该怎么办

看严重级别：严重还是不严重

严重的：紧急变更上线

不严重：修复好后跟下个版本一起上线

用户会通过运维反馈到项目组这边，项目经理会根据功能模块的负责人，分给对应的开发与测试。

测试人员：编写对应的测试用例、测试环境中重现 bug、提交 bug、

交给开发进行修复、修复完成 bug、进行 bug 的复测。

如果测试环境无法重现，可以导入生产环境的包到测试环境中测试，

还是不能复现，查看生产环境的日志去定位问题。

### 3.17 如何保证质量

(1) 需求要吃透，多问，多去了解。

(2) 严格按照测试流程去执行：多考虑用户测试场景，使用测试用例设计方法，多评审。

(3) 要有良好的测试执行：要求用例执行率达到 100%，多轮测试，进行探索性测试，

需要测试之间交叉测试，用工具来管理我们的测试工作(禅道，testlink，excel，tapd)

(4) 不断的反思与提升。

### 3.18 产品是怎么上线的？

一般我们会选择晚上上线，开发测试还有产品全部到场，进行上线测试。

首先，开发将代码打包到生产环境的服务器中，如果数据表有变化，就会运行 sql 文件，

对表的一些操作，接着，我们测试就开始先测试主体业务功能以及新增的功能模块；

测试通过之后，我们会在界面上把上线测试的数据删除，正常上线。

如果发现 bug，开发人员当场修复 bug，修复成功之后我们测试再复测，通过就可以正常上线

如果发现了 bug 开发人员在上线规定时间之前都还没有修复好的话，就看问题的严重性，

如果严重就延期上线，如果我们是迭代版本的话我们还需要版本回滚。

如果不严重，产品跟客户觉得可以上线，就正常上线。

## 四、项目

### 4.1 简单介绍下最近做过的项目

根据自己的项目整理完成，要点：

1) 项目背景、业务、需求、核心业务的流程

2) 项目架构，B/S 还是 C/S，数据库用的什么？中间件用的什么？后台什么语言开发的？

是否有做 App 端，是否有 H5 是否开发小程序等等。

3) 项目前端有哪些功能模块，后台有哪些功能模块？

## 4.2 拿一个你所负责的模块，讲下具体怎么测的？

根据自己的项目整理完成，核心要点：

- 1) 拿一个你负责过的模块，核心业务模块讲解
- 2) 业务流程是怎样的，需求怎么样，有什么规则没，规则简单介绍
- 3) 你是如何分析的，讲明分析思路，测试点，主要怎么考虑测试的，主要核心测试重点在哪里，用了什么测试方法等等。

## 4.3 你在这个项目里面主要做了些什么工作？

- 1) 在这个项目中，主要是以功能测试跟后台接口测试为主，主要参加了需求评审会议，用例的编写，参与用例的评审，执行测试。
- 2) 协助开发定位问题解决发现的 bug，编写测试报告，协助上线。
- 3) 另外就是做了 APP 的一些相关项测试，像兼容性测试、稳定性测试、安装卸载版本覆盖测试和 app 性能都是有做过的，例外后期有做过接口自动化等。主要就是做了这些工作。  
[这个具体根据你自己的简历上写的来说]

## 4.4 你们项目组有多少人、开发多少个、测试多少个？

[公司具体人数，可以不太清楚，项目组多少一定清楚]  
[这个一定要根据自己的简历项目大小来说，不能乱说]

产品 1、项目 1 个、架构师 1 个、前端 3 个、后端 5 个、ios1 个、Android 1 个、  
测试 3 个(测试主管，核心测试人员)、运维 1 个、UI 一个

## 4.5 测试人员怎么分工的？

- 1) 我们测试组 3 人，1 个测试组长，2 个测试，一般都是根据需求的复杂程度大小来，尽量是自己熟悉哪个版块的就继续做那个版块。
- 1) 比如：我这边主要是负责前端大部分的功能模块，还有接口测试跟 ui 自动化测试，另一个同事主要是功能测试这边，组长这边也负责一些功能测试，包括一些性能跟安全测试。
- 2) 其实测试工作也划分的没有那么细，后期我们也会做交叉测试，相互测试功能，性能跟安全测试我也会参与一下。

## 4.6 项目的迭代周期？多久一迭代？一个版本你们发现多少 bug

[切记工具自己所选择的项目来回答]

我们公司是这样的，迭代还是蛮快的，一般是两个星期一个迭代，迭代测试两轮。Bug 的话不一定哦，关键还得看开发，哈哈，开发的版本质量好的话，BUG 就会少些，整个版本比较好的情况下大概也就

二十来个 BUG，当然如果遇到开发是个新手，那么找到 60-70 个也是很常见的，比如之前的那个金融项目，足足发现了 72 个 BUG，这样的情况下追踪 BUG 的工作量都比较大，如果是版本迭代的话，那么基本就不会出现多少 BUG 了。

## 参考答案 2

因为我们项目的用户活动和三方合作平台比较多，一般半个月或者 1 个月肯定会有一个迭代版本，假如用户或者合作方突然有很紧急的需求，那一般老大他们会向上发邮件和 OA 呈批给（产品经理，项目经理），如果通过了就会马上加急处理这个需求，测试完成直接上线。

现在都是维护为主，但新需求也不断有，一般一个版本上百个 bug 是有的。

### 4.7 你们整个项目写了多少用例，你负责的模块大概写了多少用例？

[切记已根据自己的项目及负责的模块来]

答：这个得根据项目的复杂程度，我们最近做的这个也还好，整个项目写了大概 2 千 3 百多条（有点多了），我负责的模块就写了一千多条（你要思考，你负责了哪些模块，大概评估下，不要乱喊）。

总结注意点：没有标准答案，先说你的前置条件，再说数据，只要你前置条件和数据匹配即可。

### 4.8 最近的版本写了多少用例？

（总结注意点：没有标准答案，先说你的前置条件，再说数据，只要你前置条件和数据匹配即可

特别注意：你如果是半个月的版本，一般给你两天写用例，你自己评估下写多少。

半个月的版本：1-2 天需求分析，1-2 天写用例，1 天评审用例，其余的时间就是执行回归 bug，编写测试报告）

最近的版本因为没有特别的用户活动，产品那边也没有给特别大的改动需求，我负责的有 5 个模块吧，大概有 180 多条用例。

### 4.9 你的需求分析一般几天，用例大概写了多长时间？执行了多长时间？

如果按照 2 周一个版本来算的话，我们需求分析一般是由产品 SE 先组织我们开会，讲清新版本需求，然后我们再花 1 天到 1 天半时间去详细分析需求，另外有 2 天左右时间来写用例，写完用例会进行用例评审，后面的时间基本就是在执行用例，提 bug，并跟进 bug 修复问题。

### 4.10 在 uat 测试的时候，突然客户临时要大量的数据

**备注说明：**uat：测试人员提供用例，uat 环境已搭建好，他就开始来执行，如果发现问题，

需要协助，谁负责这个需求，就找对应的人，发现 bug，提交到 uat 版本里面，修复完了，

客户需要回归验证的，我们公司只是辅助他去执行测试。

**答案：**

看他需要的数据能不能从上个版本，或者生产环境导入数据进来测试，如果没有，我们看能不能批量修改数据去测试，如果不行，我们只能通过存储过程造数据了。

#### 4.11 发现哪些映像比较深刻 bug/经典 bug?

根据自己的项目来准备，核心要点：

- 1) 有哪些经典或者说影响比较深刻的 Bug，最好是与业务相关的 Bug，不要举例说前端的 Bug
- 2) 具体怎么分析，讲明你的分析过程。如何定位的.....

比如:业务逻辑漏洞;

支付功能:

- 1) 商品选择支付的时候，实际已扣款成功，但是用户后台显示该商品没有付款，导致不能使用该商品提供的服务。
- 2) 商品所显示的价格是 x 元，但是实际支付的时候显示和扣款的价格是 y 元 ( $x \neq y$ )

找密码流程:

按照常规操作，会直接跳跃了某个必须的流程(流程缺失)，但是通过 url 修改参数又可以访问到该流程，存在安全和逻辑漏洞。

安全漏洞:

- 1) 登录账户: 退出 or 注销之后，浏览器返回键回退之后又可以回到已登录的页面继续操作，识别用户身份的信息并没有失效，用登录后才能访问的 url 直接访问也可以登录，安全漏洞。
- 2) 搜索功能: 前端页面的搜索输入框中输入特殊敏感符号  
(如 `script> alert(document.cookie)</script>`)，直接搜索后有可能把当前登录账户的 cookie 信息直接以弹框的形式暴露出来。
- 3) 新增功能: 一开始没有限制字符的类型和数量，当输入特殊符号、超长的字符，提交后直接抛出包含有 INSERT INTO 的完整 SQL 语句。
- 4) 前端搜索: 以敏感字符直接搜索后，客户端和服务端都没有任何字符过滤 or 转义处理，直接把数据库和网站服务器的名称、版本暴露。

数据调用/加载异常:

- 1) 翻页功能: 有时候会出现前面几页翻页和数据显示都是正常的，往后再翻页就会出现翻页不了 or 加载的数据异常。
- 2) 前端页面有几级菜单的情况下，程序都已经调用过，第一级是正常展示的，但是第二级、三级有可能被折叠而没有显示在浏览器显示。
- 3) 定位到某个导航主题，调用的数据并不是该主题分类的数据，而是调用成了其他分类的数据。

不可逆操作，导致流程受阻:

- 1) APP 测试, or H5 页面测试, 触发某个操作, 比如手机触屏下滑刷新页面, 不能恢复到操作前的正常页面。
- 2) 输入某个异常值提交之后, 程序没有相关的处理机制, 导致页面保存, 没法继续进行其他操作。
- 3) 登录方式切换: 登录时有几种不同的方式, 如: 密码登录&短信验证码登录, 但同一时刻默认只能显示一种登录方式, 当从密码登录界面点击短信登录切换到短信验证码登录界面之后, 没有切换返回密码登录界面的功能。
- 4) 删除异常: 正常情况下可以从列表中删除记录, 但是若先对列表记录执行了搜索功能之后, 再次删除的时候可能出现删除无响应而删除不了数据。
- 5) 弹框阻止: 当触发某个操作, 如“保存”、提交” or 某个开关按钮, 界面中弹单出一个提示框, 此提示框不管怎么操作都无法关闭, 直接阻碍了页面上其他功能的操作。

附件上传时, 未控制格式尺寸和容量大小, 系统处理出现异常:

- 1) 文件上传功能: 没有限制上传的文件格式、尺寸和大小, 当上传非常规文件(如 js 文件)、大容量文件(如: 图片大小>20M), 较大分辨率(如: 1600×1200), 服务器没有相应的异常处理机制, 导致网站出现持续长时间的卡顿, 影响后续操作。
- 2) 上传的是非常规的文件, 如 js 格式文件, 程序无相关控制, 直接将 js 文件上传到数据库, 前端页面访问时若不能解析则出现异常页面。

缺少非空判断, 服务器报 500 错误:

- 1) 编辑包含多个字段的页面时, 有一些字段在程序中控制是必填的(事先未知), 但是没有任何说明提示, 当不填写这些字段, 直接保存时会出现服务器异常页面, 报 500 状态错误。(特别是在管理后台容易出现此场景)
- 2) 在形如以下结构的 if 函数中, 关系表达式的条件没有对某个变量(该变量因代码疏漏未作初始化赋值)进行非空判断, 就直接执行语句体, 程序已空值 null 进行参与运算而出现异常, 如 500 错误 if(关系表达式)样式导致异常。
- 3) 某个功能(如: 金额输入和统计)在 A 页面程序限制只能输入正整数, 而在 B 页面却没有相应控制, 若不小心在 B 页面输入了非正整数, 比方小数, A 和 B 的数据分别传递到同一个 C 页面时数据处理会出现异常。
- 4) 文章上传/图片上传: 超长字符的文章内容 or 较大尺寸的图片上传, 程序没有进行相关的压缩和截取, 直接完全调取到前端页面, 导致浏览样式异常,

App 测试过程中常 Bug: <https://www.cnblogs.com/123456ww/12198075.html>

[经典 bug: 前端申请借款中, 用户没有信用额度或者借款金额超过了用户信用额度但是却能成功提

交审核]

[发现途径：我是在模拟借款人，借款金额提示我的可用额度为 0，但是我输入 5000 的借款金额点击提交审核提示我提交成功，等待审核]

[解决：首先我去数据库查找到对应的表，比对我的信用额度跟界面显示的数据是否一样，一样我就把数据库的记录，填写的借款信息和我借款成功的界面显示截图都保存好。之后提交这个 bug，开发人员通过修改代码，我再复测，有没有重现 bug]

[还有一个就是在借款流程中，我们通过修改数据库中的数据，把借款时间修改了，制造出一个逾期未还款的数据，结果显示还款的金额比借款金额还少，而且管理要收得特别高，存在不合理性]

[还有一个是在产品上线后，运维人员在统计数据时发现少了一条数据，我们去数据库检查发现 0 分 0 秒的数据没有统计，后来开发人员修改了代码之后就解决了]

- 1) 服务费计算错误，计算公式开发这边写错，本来是利息 0.3，写成 003，开发修复 bug。
- 2) 退出用户，后退还可以进入到原来的登录完成操作后的界面，原始，退出用户，没有删除用户对应的 session，导致后退完成后，用户用户 cookie 可以进行操作。
- 3) 重新选择下拉框，输入信息全部清空，原因，修改类型，重新刷新界面，输入数据，并没有保存缓存里面，导致一刷新，原来信息没有，解决，开发选择不同借款类型，不再进行刷新。
- 4) 借款标题，输入 xss 攻击代码，导致接口所有的数据不存在显示，因为 xss 脚本，当然代码处理，开发这边进行转义字符串。
- 5) 谷歌浏览器登录不成功，显示验证码。

#### 4.12 每个阶段测试开发在干嘛（比如你写用例的时候开发在干嘛？）

- 1) 需求阶段，大家都在了解需求
- 2) 测试准备，  
测试编写用例，开发做概要设计，详细设计，然后就是编写代码，编写接口文档，设计文档。
- 3) 测试执行阶段，  
测试人员执行用例，发现 bug、提交 bug、开发修复 bug(开发还有可能在开发未完成的功能)

#### 4.13 你们公司是否敏捷开发

可以说是，也可以说不是。[具体看你了不了解敏捷开发模式]

[问了我有没有做过敏捷测试]

**扩展知识储备：**

##### 1、什么是敏捷开发

敏捷开发以用户的需求进化为核心，采用迭代、循序渐进的方法进行软件开发。

在敏捷开发中，软件项目在构建初期被切分成多个子项目，各个子项目的成果都经过测试，具语可视、可集成和可运行使用的特征，换言之，就是把一个大项目分为多个相互联系，但也可独立运行的小项目，并分别完成，在此过程中软件一直处于可使用状态。

## 2、敏捷开发优缺点：

### 特点：

- 1、能适应快速的客户需求变化，快速的交付，注重与客户的沟通。最优先要做的是通过尽早的、持续的交付有价值的软件，把项目拆分成各个小的子项目，快速开发快速交付，有问题及时调整，适合高风险项目。
- 2、交付周期短，交付的时间间隔越短越好，一周一个迭迭代，甚至有时候一周多个迭代，不过每个迭代版本的需求不会太多，注重项目持续迭代开发交付。
- 3、整个项目开发期间，业务人员和开发人员必须天天都在一起工作，团队规模不能太大，团队间强调面对面的交谈。
- 4、更关注可交付可以使用的软件，而非文档。
- 5、对团队技术要求高，能快速适应客户对需求的变化。
- 6、敏捷团队只专注于开发项目中当前最需要的、最具价值的部分。这样能很快地投入开发，另外，较短的迭代周期使团队成员能迅速进入开发状态。

### 优点：

- 1、敏捷开发的高适应性，以人为本的特性，适应客户的更快需求变化，更快的交付成果。
2. 更加的灵活并且更加充分的利用了每个开发者的优势，调动了每个人的工作热情。

### 缺点：

- 1、由于其项目周期很长，所以很难保证开发的人员不更换，而没有文档就会造成在交接的过程中出现很大的困难。
2. 特别项目存在新手比较多时，老员工比较累。(对开发团队人员的技术要求高)

## 3、敏捷开发流程图：

### 4.14 你这个项目做了多久？你这个项目现在的用户里有多少？活跃量多少？

时间根据简历来

比如：一年时间，金融项目：100W 用户 2W 活跃用户

## 五、测试思维

### 5.1 打电话功能怎么去测？

我们会从几个方面去测试：界面、功能、兼容性、易用性、安全、性能、异常。



- 1) 界面我们会测试下是否跟界面原型图一致，考虑浏览器不同显示比例，屏幕分辨率。
- 2) 功能：给不同人员打电话，不同号码打电话，不同运营商，测试每个按钮是否正常使用，拨打号码，是输入还是，复制过程，还是其他地方跳转过来，多次拨打电话，双卡选择不同电话卡。
- 3) 兼容性：不同手机型号，厂商，不同系统版本，屏幕大小，分辨率，内存大小
- 4) 易用性：操作是否说的越多越好

## 5.2 给你一个杯子怎么测？

功能测试：

主要关注水杯基本功能

- 1、水杯是否可以正常装水
- 2、水杯是否可以正常喝水
- 3、水杯是否有盖子，盖子是否可以正常盖住
- 4、水杯是否有保温功能，保温功能是否正常保温
- 5、水杯是否会漏水，盖住盖子拧紧后是否会漏水

界面测试：

主要关注水杯外观、颜色、设计等方面

- 1、外观是否完整
- 2、外观是否舒适
- 3、颜色搭配及使用是否让人感到舒适
- 4、杯子外观大小是否适中
- 5、杯子是否有图案，图案是否易磨损

易用性测试：

主要关注水杯使用是否方便

- 1、水杯喝水时否方便
- 2、水杯拿起放下是否方便，这里会行注到水杯形状的测试
- 3、水杯装水是否方便
- 4、水杯携带是否方方便
- 5、水杯是否有防清功能
- 6、水杯装有低温或者高温水时，是否会让手感到不适

性能测试：

- 1、水杯装满水时，是否会露出来
- 2、水杯最大使用次数

- 3、水杯的保温性是否达到要求
- 4、水杯的耐寒性是否达到要求
- 5、水杯的耐热性是否达到要求
- 6、水杯掉落时时，是否可以正常使用
- 7、水杯长时间放置时，是否会发生泄露

兼容性测试：

主要关注水杯是否可以装其他液体，如果汁、汽油、酒精等

可移植性测试：

主要关注水杯放置环境等

- 1、将水杯放在常温环境中，使用是否正常
- 2、将水杯放在零下的环境中，使用是否正常
- 3、将水杯放在高于正常温度的环境中，使用是否正常

安全性测试：

主要关注水杯外观和各种异常条件下是否释放有毒物质等

- 1、当水杯装满热水时，水杯是否会烫手
- 2、当水杯装上水后，是否会产生有毒物质
- 3、把水杯放在零下环境时，是否会产生有毒物质
- 4、把水杯放在高温环境时，是否会产生有毒物质

### 5.3 图像上传功能的测试点

1. 检查图片上传路径
2. 检查图像上传和修改功能
3. 检查各种扩展图像文件的上传（例如 JPEG、PNG、BMP 等）
4. 检查文件名中含有空格或其他可用特殊字符的图片的上传
5. 检查重复名称图片上传
6. 图片尺寸大于最大允许值，上传时应该显示适当的错误消息
7. 检查上传的图片文件类型外的其它文件时（例如 txt、doc、pdf、exe 等等），应该显示适当的错误消息。
8. 检查如果上传的图片满足指定的高度和宽度（如果有定义的话）则可以成功上传，否则不能上传。
9. 上传大尺寸图片时应显示上传进度条
10. 检查上传过程中的取消按钮是否有效
11. 检查文件选择对话框中的文件列表是否只显示支持文件类型

12. 检查上传多个图像的功能
13. 上传后检查图像质量，图像质量不应该改变
14. 检查用户是否能够使用/查看上传的图像

#### 5.4 搜索框的测试

- 1) 搜索按钮功能是否能够实现，验证搜索框的功能是否与需求一致
- 2) 点搜索后，原先的搜索条件是否清空。
- 3) 查看比较长的名称是否能查到输入过长查询数据，看其有没判断，报错系统是否会截取允许的长度来检索结果。
- 4) 是否有忽略空格的功能，需要有忽略前置空格和后置空格的功能，但不能把中间空格忽略
- 5) 不输入任何内容点击搜索看查询的结果
- 6) 查看搜索框内的默认内容是否与设置的一致，焦点放置搜索框中，搜索框默认内容是否自动被清空
- 7) 输入系统中存在的与之匹配的条件看其的查询后数据的完整性显示记录条数正确、文字折行显示正确、页面布局美观、列标题项、列显示内容、排序方式符合需求定义。
- 8) 组合中文和各种特殊符号输入查看能否正确搜索到合的内容
- 9) 输入系统中不存在的与之匹配的条件，是否搜索出信息或者给予提示信息
- 10) 使用复制粘贴，测试搜索框是否能执行
- 11) 注意在光标停留的地方输入信息时，光标和所输入的信息会否跳到别的地方
- 12) 反复输入相同的数据(5次以上)看是否报错
- 13) 敏感词汇，提示用户为敏感词汇

{

    语句提 1;

    }else{

        语句提 2

}

3. 某个被调用的方法中缺少某些参数的定义，在不知情的情况下，直接调用时传递了未定义过的参数 or 类型不匹配的参数到该方法，如果对应网站是处理批量的业务，则可能会导致大面积的 500 异常页面，对网站正常业务和 SEO 排名损失风险比较大。

4. 新增、编辑->保存，对所提交的字段有的未作非空限制，可以直接保存成功，保存后以空内容展示，可能存在不确定性，比如操作已保存成功的空记录时，是否会影响其他正常添加的记录，是相互独立的，还是会牵连到其他所有的类型。

服务器配置错误(漏配 or 错配)，更新后出现 500 or 404:

1. 服务器配置文件，如 web.config 中把前端访问的 url 地址写错，直接发布更新之后，前端页面访问可能会出现 404 错误。
2. 程序代码中的某些逻辑错误和服务器配置相冲突时，前端页面触发某些特定按钮 or 页面可能会出现 500 错误。

数据传递过程无控制，导致数据输出到界面功能异常 or 样式变形：

1. 搜索功能：有的页面本身有回显所搜索关键词的功能，搜索输入框填写的 keywords 字符较长（如：100 字符），直接搜索后这些长字符显示在页面中，使得页面原来的样式变形，甚至有的功能按钮被挤到页面之外而不能使用。
  2. 新增功能：对于新增字段的长度没有任何限制，超长字符新增可以保存成功，回到列表页也没有对显示的字符长度进行控制，所有字符长度都展示在列表，挤压其他字段的
- 14) 不同搜所的条件之间来回选择，查看是否出现页面错误
  - 15) 测试多个搜所条件时，要注意搜所条件的组合测试，可能不同组合的测试会报错
  - 16) 点击搜索框，看能否在搜索栏下方显示提供设置好的最近热门搜索词，点击任一可以直达搜索结果页
  - 17) 点击搜索框时，到有搜所历史时，能显示历史搜所内容，历史搜所内容应从上到下按时间排序，点击清空历史清空所有搜索记录
  - 18) 直看搜索框最大输入字符数

## 六、Linux

### 6.1 Linux 系统你是怎么用的？

[在测试 1、执行的过程中，我们发现的 bug，有时候需要定位 bug，协助开发修复 bug 时需要在 linux 里通过命令 tail-200 或 tail-500 查看当天的日志的后面多少行或者前面多少行定位 bug 或者通过 tail -f 来查看日志里的关键字 exception(异常) error(错误)。

[后台程序运行久了会对系统造成卡顿等诸多隐患或我们做性能测试的时候我们都会通过 linux 的命令 Ps -ef 显示所有进程)、top(监控程序执行状况)、free -m 显示内存使用情况)

来查看系统资源如果服务器出现故障时我们也会用 (service httpd status) 看下服务器是否启动，用 ps -ef|grep httpd 查看 apache 进程是否启动，用 ps -ef|grep java 查看 jdk 进程是否启动如果服务器起不来，常见的问题有端口可能被占用，用 netstat -an|grep 8080 查看端口是否已被占用。]

[搭建测试环境的时候我们是在 linux 下进行的，搭建 LAMP 时在线用命令 yum install

安装 apache, php 以及 mysql;或通过 xshell 来导入需要的环境包来搭建 LTMJ(Tomcat、Mysql、jdk)

## 6.2 Linux 是什么工具连接的，Linux 版本是多少

Xshell、CRT、SSH 用的 ssh 协议连接，端口是 22

传输文件用 xftp 工具，占用的端口是 21

Linux 版本 centos6.5 版本 64 位

## 6.3 日志具体是怎么看的

1、我们根据日志目录找到对应的日志文件，用 `tail -200`，或者 `tail-500` 查看文件内容也可以重定向导出查看。

如果是系统出现了异常导致的错误，我们跟去查找关键字，比如说 `error` 或者 `exception` 等

如果是逻辑错误，会把操作对应时间的日志跟对应开发一起去定位 bug

## 6.4 监控资源命令用到哪些

查看进程 `ps -ef` 过滤添加 `grep` 来着

杀掉进程 `kill` 强制杀掉 `-9`

监控资源 `top` `vmstat`

磁盘 `df -h`

内存 `free -m`

## 6.5 nmon 工具是如何使用的

1、下载安装包()

2、安装(不需要安装-解压即可) `nmon_linux_14i.tar.gz`

1) 把文件传输到 linux 服务器

2) 解压 xftp

```
tar -zxvf nmon_linux_14i.tar.gz
```

3) 解压文件中，找到你系统版本对应的运行文件

比如：你们的操作系统 centos6.5 64 位系统，`./nmon_x86_64_centos6` 文件

4) 运行对应的监控资源的文件

```
./nmon_x86_64_centos6
```

按字母 `c` 查看 `cpu`，`m` 查看内存，`n` 查看网络，`q` 退出

3、运行命令把数据保存到文档中

```
./nmon_x86_64_sles11 -s1 -c350 -f -m /home/
```

`-s1` 每 1 秒采集一次数据

`-c350` 采集 350 次，即为采集十分钟的数据。

-f 生成的数据文件名中包含文件创建的时间

-m 生成的数据文件的存放目录

这样就会生成一个 nmon 文件，并每十秒更新一次，直到分钟后。

生成的文件名如：\_090824\_1306nmon," " 是这台主机的主机名， nmon -h 查看更多帮助信息。

4、把生成 nmon 工具，导出到 windows

5、用 office 运行分析工具

6、生成一个 xlsx 文件

## 七、数据库

### 7.1 你们数据库怎么用

[原来我们数据库用的比较多的，就是数据结果检查，测试一些数据准备，性能测试造大量数据。]

[测试执行到的结果，我们需要通过 sql 语句 select 来查找数据库对应的表，看看数据库信息跟我们执行的结果是否一致，比如：生成申请借款后，我们会去数据库里面去检查下,数据库中数据是否跟申请订单数据一致。]

[我们在测试执行时需要做一些测试数据准备，我们就用 insert into 输入数据或(者 update set 修改数据)，我们需要到数据库查看有没有相关记录保存，保存的数据跟我们输入或者修改的记录是否一致;比如：原来我们一个初审功能里面有个分页功能，测试分页功能，需要 100 条数据，我们就通过数据库操作添加 100，可以用 insert into。也可以用脚本实现，或者存储过程]

[还有在做性能测试时，模拟用户场景时需要用到大量的数据，这时就需要我们到数据库中制造大量的数据出来。比如说，测试充值，需要大量用户数据，充值表中大量数据，比如 10W 条数据，我们就用存储过程去造。]

### 7.2 存储过程是怎么编写的

```
delimiter //  
create procedure 存储过程名(n int)  
BEGIN  
declare i int default 0;  
while i <= n do  
Insert into 表名 values(值 1, 值 2...)  
set i=i+1;  
end while;  
end //
```

delimiter;

cal 存储过程名(数据量(n));

### 7.3 常见的关系型数据库有哪些

mysql、SQL Server、Oracle、Sybase、DB2 等

MySQL 是开源免费的;

SQL Server 是由微软公司开发的关系型数据库管理系统,一般用于 Web 上存储数据;

Oracle 数据的大量性数据的保存的持久性;

### 7.4 你们用的什么数据库连接工具

Navicat, 数据库版本 mysql 5.6, 端口默认是 3306

### 7.5 左连接与右连接有什么区别

左连接: 以左边的表(employ)为主, 显示左边表列的全部数据, 如果右边表没有对应的数据, 则为 NULL

右连接: 以右边的表(student)为主, 显示右边表列的全部数据, 如果左边表没有对应的数据, 则为 NULL

### 7.6 索引有哪些, 如何建立索引, 索引的优缺点

MySQL 索引的建立对手 MySQL 的高效行是很重要的, 索引可以大大提高 MySQL 的检索速度

缺点: 虽然索引大大提高了查询速度, 同时却会降低更新表的速度, 如对表进行 INSERT、UPDATE 和 DELETE, 因为更新表时, MySQL 不仅要保存数据, 还要保存一下索引文件, 建立索引会占用磁盘空间的索引文件。

索引分单列索引和组合索引, 单列索引, 即一个索引只包含单个列, 一个表可以有多个单列索引, 但这不是组合索引, 组合索引, 即一个索引包含多个列。

主键索引 PRIMARY KEY, 唯一索引 UNIQUE, 普通索引 INDEX

组合索引 INDEX, 全文索引 FULLTEXT

## 八、抓包与网络协议

### 8.1 抓包工具怎么用

我原来的公司对于抓包这块, 在 App 的测试用得比较多。我们会使用 fiddler 抓取数据检查结果, 定位问题, 测试安全, 制造弱网环境;

如: 抓取数据通过查看请求数据, 请求行, 请求报头, 请求正文, 信息是否正确去检查结果,

如果是 4 开头的话就有可能是前端问题一般我会到前端排查，以 5 开头就有可能是后端问题我就会到后端排查；如果是 200 的话，就需要检查请求行，请求报头，请求正文是否正确，如果请求错误就是前端问题，如果请求没有问题，那就是后端问题，看后端问题服务器运行日志，是否包含 exception, error 或根据时间点去看日志。

测试安全，抓取数据查看用户的敏感信息有没有进行加密显示，还有就是把发送请求的数据篡改是否成功。

弱网环境，通过 fiddler 工具选择 Customize Rules... (Ctrl+R) 调出定义脚本编辑器找到

“if (m\_SimulateModem)” 设置上行下行网速，然后把

Rules-> Performance-> Simulate Modem Speeds 选中生效

常用抓包工具有：浏览器中 F12, fiddler, Charles(青花瓷), wireshark

## 8.2 如何抓取 https 的包

1、设置 Tools=> Option=>勾选 Decrypt Https traffic=>勾选 ignore server certificate errors(unsafe)

2、打开 https 网页就可以成功抓取了

3、还可以 Fiddler 添加过滤器(Filters)：只抓取指定 ip 的数据

## 8.3 如何抓取手机的包

1、开启 Fiddler 的远程连接

Fiddler 主菜单 Tools- Options-> Connections>勾选 Allow remote computers to

2、重启 Fiddler，更新刚开启的远程配置

3、然后手机和电脑需要在同一个局域网，抓取 http 手机设置代理就可以，要抓取 https 包，手机需要安装一个 fiddler 证书

1、fiddler 工具生成一个证书，发送手机上面安装

2、通过手机浏览器打开安装证书界面 192.168.3.197: 8888

ip 地址是用 fiddler 工具的电脑的 ip 地址，fiddler 工具端口号的 8888

3、点击下载证书，会提示，输入手机锁屏密码

4、给证书命名，名字随意，其他默认就 ok

5、点击确定，安装成功，然后就可以抓取 https 的包了

## 8.4 网络协议了解多少？

原来我们用得比较多的协议是 http 和 https 以及 tcp 协议

http 和 https 都是超文本协议，浏览器发送数据请求基本用的都是他们，不同的是 https



在 http 的基础上增加了 ssl 加密协议, http 的默认端口是 80, https 的默认端口是 443, https 收费, http 免费。

tcp 协议的话, 作用在传输层, 在发送请求前会有三次握手, 是面向连接的协议, 传输过程比较可靠  
udp 协议的话, 作用在传输层, 面向非连接协议, 传输过程相对 tcp 不可靠, 传输大量数据

### 8.5 请求方式有哪些?

常用: get、post

不常用: delete、put、head、option

### 8.6 get 跟 post 请求的区别

- 1) get 请求的参数有长度限制, post 没有
- 2) get 请求参数在 url 上传输, post 的参数在请求正文中传输。post 比 get 传输更安全
- 3) get 只能接收 ascall 码参数, 而 post 没有限制
- 4) get 请求的时候, 只请求一次, 而 post 请求两次, 第一发送请求头相关信息, 第二次再发送请求正文, (只有部分浏览器 2 次请求)

### 8.7 http 跟 https 的区别

1. https 协议需要到 ca 申请证书, 一般免费证书较少, 因而需要一定费用
2. http 是超文本传输协议, 信息是明文传输 https 则是具有安全性的 ssl 加密传输协议
3. http 和 https 使用的是完全不同的连接方式, 用的端口也不一样, 前者是 80, 后者是 443
4. http 的连接很简单, 是无状态的; https 协议是由 SSL + HTTP 协议构建的可进行加密传输、身份认证的网络协议, 比 http 协议安全。

### 8.8 为什么要使用 cookie 和 session: http 是无状态协议

第一次登录, 发送用户信息给到服务器, 服务器把用户信息保存在 session 中服务器响应数据给客户端, 响应数据中有包含 session 的先关用户信息

客户端接收到服务器 session 信息, 把 session 中相关的用户信息保存在 cookie 中

第二次登录, 客户端发送请求, 并携带 cookie, 服务端可以直接验证 cookie 值, 如果用户已经登录过, 可以免登录

### 8.9 cookie 跟 session 的区别

在网站中 http 请求是无状态的, 也就是说即使第一次和服务器连接后并且登录成功后,

第二次请求服务器依然不能知道当前请求是哪个用户, cookie 的出现就是为了解决这个问题,

第一次登录后服务器返回一些数据(cookie)给浏览器, 然后浏览器保存在本地, 当该用户发送第二次

请求的时候，就会自动的把上次请求存储的 cookie 数据自动的携带给服务器，服务器通过浏览器携带的数据就能判断当前用户是哪个了。cookie 存储的数据量有限，不同的浏览器有不同的存储大小，但一般不超过 4KB，因此使用 cookie 只能存储一些小量的数据。

session 和 cookie 的作用有点类似，都是为了存储用户相关的信息，不同的是，cookie 是存储在本地浏览器，而 session 存储在服务器。存储在服务器的数据会更加的安全，不容易被窃取。但存储在服务器也有一定的弊端，就是会占用服务器的资源，但现在服务器已经发展至今，一些 session 信息还是绰绰有余的

## 8.10 post 申请方式，用 get 会报什么错误。

404 Not Found

请求失败，请求所希望得到的资源未被在服务器上发现，没有信息能够告诉用户这个状况到底是暂时的还是永久的，假如服务器知道情况的话，应当使用 410 状态码来告知旧资源因为某些内部的配置机制问题，已经永久的不可用，而且没有任何可以跳转的地址，404 这个状态码被广泛应用于当服务器不想揭示到底为何请求被拒绝或者没有其他适合的响应可用的情况下，出现这个错误的最有可能的原因是服务器端没有这个页面。

## 8.11 http 协议提交请求头内容

Accept-Charset: 浏览器能够显示的字符集

Accept- Encoding: 浏览器能够处理的压缩编码

Accept-Language: 浏览器当前设置的语言

Connection: 浏览器与服务器之间连接的类型

Cookie: 当前页面设置的任何 Cookie

Host: 发出请求的页面所在的域

Referer: 发出请求的页面的 URL

User-Agent: 浏览器的用户代理字符串

Content-Type: 请求数据的格式或者是类型

## 九、接口测试

### 9.1 接口测试怎么测

(jmeter 版本)

首先开发会给我们一个接口文档，我们根据开发给的接口文档，进行测试点的分析，主要是考虑正常场景与异常场景，正常场景，条件的组合，参数的格式校验等价边界值;异常场景，多一个参数，少一个必填参数，参数为空;接着编写测试用例，用的 Jmeter 工具去运行，创建线程组，建立 http 请求，

输入测试用例，请求参数，建立察看结果树，运行，看返回的结果，是否跟接口文档里面要求的返回结果一致，其他用例，值需要修改里面的参数，请求地址这些信息。

举例说明：（不要登录跟注册）

比如说原来我们做一个申请借款的接口，对接口进行测试分析，考虑正常场景与异常场景，正常场景，考虑不同参数组合，比如说，不同借款方式，还款期限，还款日期，借款的利率等参数组合；也要测试每个参数格式校验，异常场景，：多一个参数，少一个必填参数，比如没有借款的利率，参数为空的，比如借款标题为空，编写测试用例

在 jmeter 中执行，填写参数，更地址就 ok，发送请求

（python + request）

原来我们接口主要是用的 python + requests 去运行的

首先，开发会给我们一个接口文档，拿到接口文档后，我们就进行测试点的分析，

考虑正确场景，条件的组合，

异常场景，多一个参数，少一个参数，参数为空的情况

比如原来我们做一个生成订单的接口，考虑正常场景，异常场景

正常场景就是不同的订单类型，订单金额，能不能申请订单，每个参数的格式类型的校验，

异常场景，多一个参数，少一个必填参数的时候，还有参数为空的情况

原来我们是用 python + request 去做的接口

首先，导入 request 包

建立一个 headers，保存请求头的信息，因为订单请求方式是 post 类型，数据格式是 form 表单格式，我们把数据保存到 data 的字典里面

这个时候我们还需求登录的 cookie 值跟登录后产生的 token 值

我们会去通过动态关联去获取登录的 token 跟 cookies，

cookies 值的话，我们是直接调用登录返回的 cookies、token 值的时候，我能是通过导入 re 模块，通过正则表达式去提取

当参数，headers、cookies 输入完成以后，我们就发送请求，打印返回结果，检返回结果是否跟我们测试用例一致

当运行其他测试用例时，我们去修改 data 里面的参数就行，在发送请求

有的请求时 https 协议的时候，我们发送请求的时候还会 very= false 去忽略掉证书验

证对应多个接口调用 cookies 我们会用到 session 去保存接口发现比较多的问题，就是格式校验这块

比如说我们提交订单，订单数据没有显示，订单格式也没有显示，输入字母，汉字都可以

订单类型为空，也会生成订单成功

我觉得接口可以发现接口更多的 bug，还可以提早进行测试，提高测试的质量

## 9.2 两个接口有关联，jmeter 具体怎么做

另外两种问法：上个接口的返回值是下个接口的请求参数，这种如何处理？动态关联有没有了解过？

这个涉及到动态关联，首先要搞清楚后一个接口需要用到上一个接口的什么数据，例外要看数据是在哪里取的，是在 head 还是在 body 里，然后如果要取的数据是 json 格式我会在发请求用 json 提取器去取这个数据，如果是其他格式的就用边界提取器或正则表达式去取数据

就拿我当时做的那个下单接口来说吧，因为下单接口需要先登录，需要用到登录接口的 cookies 来做鉴权，首先就是把登录接口调试通过，然后在登录接口的 http 请求中添加一个边界值提取器或者也可以用正则表示式提取器去提取登录接口的响应头中的 cookies 值然后在下单接口中需要添加一个 http cookies 管理器，在 http cookies 管理器中引用登录接口提取出来的 cookies，这样就可以

如果是不同的线程组的话，那在登录接口中还得添加一个 Beanshell 取样器，在 Beanshell 取样器中，利用函数助手中的 SetProperty() 函数把提取出来的 cookies 设置为全局变量，然后在下单接口的 http cookies 管理器中利用函数助手中的 Property() 函数引用登录接口中设置的全局变量，这样就可以。

## 9.3 接口测试主要目的是什么？

例外两种问法：接口测试的价值，意义？为什么要做接口测试？

主要就是验证后台服务端的业务逻辑有没有问题，提高测试的效率

①越底层发现 bug，它的修复成本是越低的

②前端页面修改频繁情况下，接口测试不受页面元素改变而影响

③检查系统的安全性，前端传参不可信，比如京东购物，前端价格不可能传入-1 元，但是通过接口可以传入-1 元

④如今的系统复杂度不断上升，传统的测试方法成本急剧增加且测试效率大幅下降，接口自动化测试可以提高测试效率

⑤接口测试相对容易实现自动化持续集成，且相对 U 自动化也比较稳定，可以减少人工，回归测试人力成本与时间，缩短测试周期

## 9.4 接口测试的流程

1，首先分析开发给到的接口文档

2，接口文档分析完成，编写测试用例

3, 然后借助接口测试工具去测试执行测试用例

4, 发现 bug 提交 bug, 并跟进 bug 修复

### 9.5 接口测试和平常的 UI 测试有什么区别?

其实这两者测试的侧重点是不同的, 接口因为没有界面, 更多考虑后台服务器对请求的, 处理逻辑问题, 业务交互, 检测的是后台“容错机制”是否完整;

而 ui 更多会去关注页面展示, 数据转换, 界面排序这些功能, 当然也会后台数据处理的问题, ui 测试其实已经包含了接口测试。系统功能的用例更全面, 不仅有界面的, 也有业务功能用例, 还有其他用户场景的用例功能入口用例, 流程用例, 而接口测试主要根据各种入参场景来设置用例。

### 9.6 给你一个新的接口, 你怎么去设计用例?

首先要对于每个要测的接口都要先搞清楚这个接口的功能, 它的作用是什么, 熟悉这个业务功能需要用到什么协议, 请求方式是什么, 接口有哪些参数。对于每个参数的作用都要搞清楚, 像数的类型, 是否有约束限制, 是否为必填的, 长度, 其他的限制等等, 如果两个参数之间有关联我们还要考虑参数的组合场景, 对于参数不理解的, 一般都会跟开发沟通下, 然后考虑返回数据的类型, 返回数据中的返回码和返回信息是什么, 通过以上几个点去提炼测试点, 设计用例。

参数约束——长度、必选项、格式、数据类型

业务场景——正确的业务场景;错误的业务场景;异常场景: 服务器空间不足

组合场景——相互依赖: 手机和验证码、用户名和验证码;

相互排斥: 二选一当然还有边界值等价类等等

Jmeter 测试流程, 步骤如下:

创建 jmeter 线程组—添加 HTTP 请求—输入协议—域—端口—路径—编码—请求方式—请求参数—启动

Jmeter 测试流程:

先需求, 再根据需求写测试点转换成测试用例, 根据测试用例编写测试脚本; 执行测试脚本;

提交 BUG, 跟踪 BUG

### 9.7 接口文档主要包含哪些内容?

接口文档一般两种形式的, 要不就是 word 版本的不要不就是 htm 的形式, 具体内容

1. URL(接口地址)

2. 接口功能

3. 请求方式: post

4. 请求参数, 以及接口中每个参数的详细说明, 类型, 是否为必填, 约束条件等等

5. 响应数据及格式, 返回码, 返回码解释等等

## 9.8 你们什么时候测试接口

一般有需求就会做，后台的接口开发好，就可以开始测。例外，如果增加了新需求，也要做接口测试，还有就是开发对后台的接口做了修改，交互逻辑发生变化，我们也要重新对接口进行测试。

## 9.9 你怎么去检查，分析

我们主要是根据入参情况，去看接口的返回值，对于返回值，我主要关注的几个点：1. 状态码  
2. 提示信息 3. 返回数据的具体内容。根据接口文档的说明去检查这个 3 个点是否满足接口需求文档，  
4. 有些如果要检查数据库的，就连接数据库获取数据与返回的数据做对比。  
如果不满足就是有问题，如果满足则通过，如果有 Bug 我们会先大概分析下，是什么原因，并进行复测，如果还是有问题，提交 Bug 给开发，让开发修复，之后再回归测试

## 9.10 什么是 api 接口测试

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

## 9.11 什么情况下开展接口测试？

- 1、项目处于开发阶段
- 2、有接口需求文档，开发已完成联调，功能测试展开之前
- 3、专项测试：参数约束测试，业务场景测试，测试接口请求响应时间（性能）
- 4、版本上线前，进行整体回归测试，查看接口是否有异常(如 404 等)

## 9.12 依赖于第三方的接口如何测试

- 1、需要第三方接口的，接口文档
- 2、发送请求到第三方接口，检查第三方接口返回的数据是否正确
- 3、不正确的时候，要跟第三方接口联调，看是请求问题，还是第三方接口返回数据有误，  
这个我们公司的第三方接口，我们都是打通的，比如电商，我们通过调用微信接口等等，都是打通的，  
比如要测试下单第三方支付，我们自己开店，收款设置我们自己的账号，然后通过商品设计 1 分钱，去测试的。

如果不打通的话，基本也只能抓包，主要保证我们发送出去的数据符合需求文档就行，然后真正的上线之前，我们会在预生产环境做一个联调测试，把各自系统连在一起，做一个联调测试没有问题了  
我们就可以上线，基本就这么做的

联调测试怎么做的：

其实联调测试就是数据拉通测试，两个子系统，连在一起，形成一个完整的系统，然后从上游下数据，

下游接到数据看传过来的数据是否符合下游的系统要求然后下游做了操作，把数据返回给上游，通知上游说数据返回了，上游看返回的数据是否符合要求，如果没有问题，就这个数据就拉通成功这个都是按照用例来执行，上游和下游一起出一份用例，两边都评审通过，然后按照测试用例执行，每条用例测试通过那么联调测就完成了。

### 9.13 你们接口怎么鉴权的？

- (1)通过用户和密码，auth 鉴权
- (2)通过 cookie 和 session
- (3)通过 token
- (4)通过 sign 签名

现在 app 一般是通过 token 鉴权，有些是通过把 token 放在请求头里面，有些是通过 singn 签名这个字段放在 body 里面去鉴权的，一般的 web 是通过 session 去鉴权的

### 9.14 接口传输格式有哪些

常见的媒体格式类型如下：

text/html: HTML 格式

text/plain: 纯文本格式

text/xml: XML 格式

Image/gif: gif 图片格式

mage/jpeg: jpg 图片格式

Image/ng: png 图片格式

以 application 开头的媒体格式类型

application/xhtml + xml: XHTML 格式

application/ml: XML 数据格式

application/atom + xml: Atom XML 聚合格式

application/json: JsoN 数据格式

application/pdf: pdf 格式

application/msword: Word 文档格式

application/octet-stream: 二进制流数据(如常见的文件下载)

application/x-www-form-urlencoded: encoded: <form encType=" " >中默认的 encType, form 表单数据被编码为 key/value 格式发送到服务器(表单默认的提交数据的格式)

另外一种常见的媒体格式是上传文件之时使用的：

multipart/form-data: 需要在表单中进行文件上传时, 就需要使用该格式

## 9.15 cookie、session、token 的区别

它们都是用来做鉴权的, 区别的话, 大概是这样的

1、现在 cookie、session 一般是配合使用的用户第一次登陆时, 服务器会创建一个 session 生成一个 sessionId, sessionId 保存在 cookie 中, 然后返回到客户端, 保存在浏览器中。

客户端每次发请求都会把这个值带到服务器, 做一个鉴权和会话的跟踪, 或者时效的验证

2、token 和 cookie、session 差不多, 通过算法, 每次登陆, 会产生一串很长的随机字符串, 一般是在放在返回的 body 里面, 或者返回的头里面, 他们都是服务器产生, 带过来是要做验证和时效的验证的。一般在 app 中使用 token 比较多一点, Web 端使用 cookie、session 的鉴权方式会多一点。

## 9.16 接口测试的工具有哪些?

Fiddler 抓包工具, 也可以做接口测试

Postman 接口测试工具, 支持接口自动化测试

Wireshark 支持电脑上各种协议的抓包工具, 主要常见有 http 和 tcp 抓包

SoapUI 功能强大的接口测试工具, 性能测试, 接口自动化测试

java+HttpClient.jar java 代码实现接口自动化测试, 一般需要借助单元测试框架 JUnit 和

TestNG 接口自动化测试框架设计: java+HttpClient+TestNG

Python + requests python 代码实现接口与接口自动化测试, 测试框架: unittest, pytest,

接口测试框架设计: python+requests+unittest+HTMLTestRunner 或者 python+requests+pytest

LoadRunner 接口自动化测试, 接口性能测试(主要)

JMeter 接口测试, 接口自动化测试, 接口性能测试(主要)

Swagger 编写在线接口文档, 在线接口测试

## 十、接口自动化

### 10.1 接口自动化怎么测试

(Python+requests+pytest 版本)

原来我们接口自动化是用 python+request+pytest 执行

接口自动化其实主要就是接口测试的基础上填加了断言, 参数化, 动态关联

做接口自动化之前, 我们也会划分模块, 报告, 公共的模块, 测试数据, 测试报告, 主要的目的是为了后期方便维护

测试数据, 一般原来我们就是用的接口测试用例, 公共的模块, 主要是里面的一些公共的作, 比如说用例 excel 数据的读取



数据库的连接，还有我们封装的每个接口请求

断言的主要是获取访问接口的值判断，用的是 `assert`，参数化主要用的比较多是 excel 表格，就是测试用例数据

还有需要调用登录后的 `cookies` 跟 `token` 的时候，我们就会用到关联

比如说原来我们写的一个申请借款的接口吧

首先我们会编写测试用例，把每个用例数据保存到 excel 中

再建立一个申请借款的模块

这个时候我们去调用申请借款的功能模块，里面的参数我们是保存在 excel 表格中

我们建立发送请求，通过参数化，去读写 excel 表格中的数据，获取到返回的数据，通过 `assert` 去对应返回的数据跟用例中异常的数据。

这个时候也会做数据库断言，去连接数据库去查询数据库中时候存在查询，如果是返回结果是 json 数据格式，我们还会转化下格式后，再去断言

这个申请借款模块，也会用到登录的 `cookie` 值 `token`，我们先建立一个登录的请求，提取返回的 `cookie` 值 `token`

excel 表格多个用例，我们就用到循环去运行，读取 excel 中用例总的条数，去循环运行，这里要注意的是：

就是 excel 表格数据时是 str 我们要 `eval` 转化成字典格式

把每个接口封装好以后

我们就会调用 `pytest` 框架去运行所有 test 文件的测试用例

如果只是执行部分用例，也可以通过 `pytest` 框架来指定

然后用 `yagmail`，在 `pytest` 框架运行完 test 文件之后，发送邮件到指定邮箱。

接口自动化，我个人觉得，性价比是比较高的。

实现起来简单、维护成本低，容易提高覆盖率等特点

接口是稳定的，最多是增加一个字段或者新增接口之类的

低成本，有了相对的稳定性，不需要大量重新编写脚本，只需要基础维护包括用例的扩充就可以了执行的快，反馈的速度快

（jmeter 版本）

原来我们也做了很多接口自动化，接口自动化这块，其实原来我们也是用 `jmeter` 请求去做的，这个时候，我们也用到一些工具，http 代理，主要方便编写接口请求，通过录制就行了，我觉得接口自动化只是在接口测试中多加了一些参数化、关联、断言参数，主要是函数参数化，自定义变量参数化，文件参数化，主要文件类型 `csv` 跟 `txt`，不过原来 `csv` 文件用的比较多，还有一些数据库的参数化，

断言，主要响应代码断言，响应文档断言。

比方说，原来我们一个登录接口主要是正常场，异常场票这块，正常场景，主要是用户跟密码正确，采用数据参数化，把用户名用随机函数进行参数化，随机长度大一些，用户名不存在的情况，原来是通过文件参数化，设置参数值，密码不正确也是通过文件参数化，接口请求中 host 地址，目录地址，我们都进行数据化，自定义变量去操作，结果检中，我们主要是用断言来检查，每个请求，设置了 2 个断言，一个响应代码断言，一般是 200，响应文本断言，登录成功，返回码为 1 状态提示成功，检查是否成功，对应异常场景也是，都需要设置断言，去检查结果原来做的申请借款接口，需要登录接口 http cookie，我需要建立 2 个接口，一个登录接口，一个申请借款接口，通过正则表达式去提取登录接口返回 cookie，在申请借款请求接口，设置 http cookie 时，值为登录接口返回 cookie，还有也要考虑原来我们项目，还有 token 值，提取登录返回 token，提取，当成申请借款的请求参数，当测试场景的脚本编写完成，执行接口测试用例，我们在察看结果树中，检直，主要是看颜色这块，红色检查哪些地方失败，绿色表示通过

编写完成后，我们会把脚本添加到 jenkins 里面持续集成运行

原来我们持续集成是半个月运行一次，当然我们也可以手动构建

- 1，我们一般把写完的 jmeter 的脚本
- 2，通过 svn 把写好的脚本检入到 svn 服务器
- 3，在 jenkins 任务下，选择定时构建，或者手动构建，检查 svn 上传最新的脚本，去运行
- 一般我们项目在修改新的功能模块，上线，转测之前，都会自动去运行脚本
- 4，运行完成，我们再 jenkins 下，查看脚本运行结果

## 10.2 为什么做接口自动化？

接口自动化，我个人觉得，性价比是比较高的

实现起来简单，维护成本低，容易提高覆盖率等特点

接口是稳定的，最多是增加一个字段或者新增接口之类的

低成本，有了相对的稳定性，不需要大量重新编写脚本，只需要基础维护包括用例的扩充就可以了执行的快，反馈的速度快

## 10.3 假如公司想要做自动化，让你去做，你会从哪些方面考虑入手？

1. 测试范围
2. 时间进度
3. 人员安排
4. 框架确定
5. 环境的搭建
6. 准备好测试数据 数据驱动
7. 工程的管理后期的维护

## 10.4 你写了多少接口自动化用例

自动化用例，也没有具体数过，当时我负责的所有模块的接口的自动用例都是我这边独立完成的，

有模块的用例会多一点，有些会少一点，这具体看接口的参数有多少，参数多限制条件多的，一般用例会比较多一点，我负责的模块大概有 100 多条用例是有的！

#### 10.5 比如说你接口的请求参数需要加密处理的，你们用的是什么加密方式，你加密怎么处理的？

这个是有做过的，就拿当我们那个项目的登录接口来讲吧，那个登录的密码是需要进行加密加密之后再进行传输。这里需要问开发要加密算法，我们会把它封装成一个函数，调用这个加密函数对密码加密，之后再进行传递。我们公司的加密算法，大部分用的都是 MD5 的加密算法(base64)

#### 10.6 你查询出来返回结果是密文，密文你怎么测试

这里首先要搞清楚用的是什么加密算法，问开发要解密算法，对返回的数据进行解密解密完成之后在与预期结果对比，去进行断言

#### 10.7 http 如何进行代理录制接口

Web 端

1，浏览器设置代理就可以录制，默认 ip 为 localhost，端口 8888

手机端

1，手机设置代理就可以录制

默认 ip 为 pc 机器的 ip 地址，端口 8888

#### 10.8 jmeter 如何进行参数化，参数化类型包含哪些

用户参数自定义变量文件参数化，csv 文件或者 txt 文件  
函数助手随机函数，csvread 函数数据库参数化

#### 10.9 jmeter 中对于 json 数据如何提取信息

正则表达式提取或者 JSON Extractor 提取

#### 10.10 jmeter 中如何跨线程组传输参数

正则表达式或者边界值提取器或者 JSON Extractor 提取的值

后置处理器- beanshell 处理器

定义成全局变量

```
${_setProperty(newtoken, ${access_token},)}
```

其他线程组，引入变量值

```
${_P(newtoken,)}或者${_property(newtoken,)}
```

## 10.11 jmeter 如何进行断言

### 1, 响应断言

添加响应断言: 添加-》断言-》响应断言

apply to:是应用范围, 设定匹配的范围

Main sample and sub-samples:匹配范围为当前父取样器, 及子取样器

Main sample only:仅当前父取样器

Sub samples only:仅子取样器

Meter Variable:变量值进行匹配

要测试的响应文字:针对响应数据不同部分进行匹配

(1)响应文本: 响应服务器返回的文本内容, http 协议排除 header 部分

(2)响应代码: 匹配响应代码, 比如 http 请求中 200 代表成功

(3)响应信息: 匹配响应信息, 处理成功返回成功”或者“ok”字样

(4) Response Header 匹配响应头中的信息

匹配规则:

包括: 响应内容包括需要匹配的内容就算成功

匹配: 响应内容要完全匹配内容, 不区分大小写

equals: 完全相等, 区分大小写

substring: 响应内容包括匹配内容即为成功

可以通过添加断言结果来查看断言的执行情况

执行结果:

如果接口响应数据可以与断言匹配上, 则测试用例通过, 否则不通过

可以通过断言结果, 查看断言执行情况。

### 2, 大小断言

### 3, 持续时间断言

## 10.12 jmeter 如间在 cmd 命令下运行

Jmeter -n -t 文件路径\fw-zhuce.jmx -l result.jtl -e -o E:\resultreport

讲解: 非 GUI 界面, 压测参数讲解

-h 帮助

-n 非 GUI 模式

-t 指定要运行的 JMeter 测试脚本文件

-l 记录结果的文件每次运行之前, (要确保之前没有运行过即 xxx.jtl 不存在, 不然报错)

- r Jmeter.properties 文件中指定的所有远程服务器
- e 脚本运行结束后生成 html 报告
- o 用于存放 html 报告的目录(目录要为空，不然报错)
- R 表示选择执行=远程启动

XXX.XXX. XXX. XXX: 5174 , XXX. XXX. XXX. XXX: 5172

官方配置文件地址 <http://jmeter.apache.org/usermanual/get-started.html>

### 10.13 jmeter 运行完成后如何去自动发送邮件?

#### (1) 监听器中添加-邮件观察仪

文件名-运行完成，保存运行结果的位置

from 邮件的发送人

iszl8lxiongmao@126.com

addressee 邮件的接收人(多个人用逗号隔开)

success subject 运行成功，发送邮件标题 success limit 大于运行请求成功的次数

failure subject 运行失败，发送邮件标题 failure limit 大于运行请求失败的次数

host 邮件服务器地址 smtp.126cm

login 邮件服务器登录用户名密码(授权码)

1、文件名：只需要给出路径和保存的文件名称即可，给定之后将会把测试结果的数据写入到文件中

注：它不会将此文件已附件的形式在邮件中，只是将测试结果写入到了定的此目录文件中，

如果你运行完脚本，直接在此路径下打开此文件就可以看到运行结果

2、Success Limit 与 Failure Limit：当成功数与失败数为几时进行邮件的发送(注意：此处是大于给定的数值，不是等于)，我写的 1，则失败 2 次后将发送邮件通知我，

3、当测试结果 100%成功时则不会发送邮件

4、写代码 java 编写 beanshell 后置处理器

### 10.14 pytest 如何做断言?

用 assert 断言

1，断言返回的结果

2，进行逻辑检查，检查数据库产生的数据

### 10.15 patent 中如何去调取其他用例中返回的参数?

把返回的值定义成全局变量

global a\_id #定义成一个全局变量

```
a_id = incharge_id
```

## 10.16 你们做接口自动化，用例数据是怎么组织，管理的？

用例数据这块，当时公司要求使用 excel 表格来进行管理，其实这里主要也是为了实现数据与脚本的分离，提高整个工程后期的维护与优化，这里把数据封装到 excel 表格之后。

我们在脚本中通过调用封装好的读取 excel 表格的数据函数，对 excel 表格中的用例数据，我们是这么组织的，会有以下几个字段像用例标题，请求地址，请求方式，请求头，请求参数，响应结果，这个几个部分，对于请求头跟请求参数，因为脚本中发请求都是通过组装成字典的形式来发送的所以这里我们也是通过类似于字典的形式文本格式来进行组织，主要就是方便后期脚本的提取与引用其实我觉得，这样去处理的好处就是，后期如果用例数据有变动，或者需要增加或删除部分用例直接针对 excel 表格数据进行操作就可以了，不需要改动脚本这也就方便整个项目工程的管理与维护了。

## 10.17 requests 中如何进行动态关联

- 1，如果返回的是 cookies 值，可以直接返回接口的 `r.cookies`
- 2，返回的是 str 类型数据，可以导入 re 模块进行正则表达式提取返回数据格式是 json 格式，导入 json，把 json 数据格式转化 python 对象

|                         |                             |
|-------------------------|-----------------------------|
| <code>json.dumps</code> | 将 Python 对象编码成 JSON 字符串     |
| <code>json.loads</code> | 将已编码的 JSON 字符串解码为 Python 对象 |

## 10.18 你们 python 接口自动化怎么去处理 cookie， session 的？

对于 cookie，session 的处理一般有三种方式：

第一种就是先获取登录请求的 cookie 值，然后发送其他请求的时候，在 requests 提供的两个方法 get 或 post 方法中有一个 cookies 参数，我们可以通过这个参数来传递 cookies 值  
第二种就是通过订制请求头，然后把获取到的 cookies 放在请求头中，通过请求头来进行传递  
第三种就是通过创建一个 session 会话对象，后期所有的请求发送都通过调用这个 session 会话对象来进行发请求，如果是登录请求，它会自动保存 cookies 值，然后其他需要用到 cookies 值的请求，也通过 session 对象来发送，它会自动把 cookies 发送出去，对于 cookies，session 的处理，我们差不多都是通过以上三种方式来实现的

# 十一、UI 自动化

## 11.1 ui 自动化怎么测试

(1)我们做 U 自动化测试使用 Python+ selenium+ unittest 工具

主要对自动化进行规划，把数据、代码、资源、公共方法进行分离，方便后续维护；

具体是通过元素定位、断言、参数化和封装来实现；

其中元素定位的方法，主要是 css 定位，也会用一些 xpath 定位，还有 id、name、class 等断言，主要是用 Assert 来判断实际结果是否与预期结果相符；

参数化，可以用列表、字典、导入 excel 表格库(xlrd)对数据进行参数化；

对于封装，一般我会对元素还有公共的模块进行封装。

比方说：原来我们有一个申请借款的功能模块

首先我们先进行自动化框架分类，分成测试数据、需要定位的元素、公共方法(比如说，一些数据的操作、excel 操作，我们会封装成一个公共方法)。先采用 css 定位，把申请借款所对应的元素定位出来，里面有很多内嵌页面，还有下拉框元素的定位操作，对应申请借款的数据，我们都会保存在 excel 表中，从 excel 表格中去获取数据。申请完成以后，我们会设置一个断言，还有进行数据库的检查。我们会把元素封装到资源库，还有就是模块的封装，申请借款会用到登录模块，我们首先会封装好登录功能。借款申请完成后，会从数据库中删除相关的数据实现数据闭环，再关闭界面。

运行自动化脚本，我们是用 unittest 框架去运行，编写测试集，一般我们是编写 unittest 脚本，去执行所有测试用例。

我觉得自动化中最难的在于元素的定位、内嵌界面、时间控件，这些需要一些 js 操作去除元素属性，进行操作。

自动化中发现的问题：

在申请借款中，开发人员写了调试代码的弹出框，没有删除，导致我们自动化脚本跑不通充值过程中，

添加新的支付方式，开发修改了代码，导致原有充值失败，这些都是我们原来自动化发现的问题。原来也用了一些 Robotframework 的自动化测试框架，这个比较简单一些，因为部分关键字、框架已经封装好了，只需要调用就行，但是不好的地方，就是扩展性会差一些。

编写完成以后，可以用 Jenkins 持续集成去定时跑自动化测试用例，再去查看运行日志。

总的来说 RF( Robot Framework)比较简单方便上手比较快，如果公司对自动化要求不高的话可以选择 RF 框架来做，如果比较高的话就选用 unittest 框架来做。

(2)原来我们自动化测试主要是用的 python+ selenium+ pytest 框架做的

主要对自动化进行规划，把数据、代码、资源、公共方法进行分离，方便后续维护；

自动化测试其实主要通过几个方面元素定位，参数化，断言，模块的封装，数据闭环等来实现的

其中元素定位主要是 cs 定位也会用一些 xpath, id, name, classname, ink 等等定位，还有 js 定位和 jQuery 定位都会用到；

参数化主要是用列表、字典、以及导入 excel 表格库(xlrd)对数据进行参数化;

断言: 主要是用 assert 断言来判断实际结果是否与预期结果相符;

对于封装, 我主要会对界面上的元素, 一些公共方法(数据库操作, 读取 excel 数据公共方法等), 还有测试数据(变量, 字典, 列表 excel 表等)和一些元素操作, 测试用例等等进行封装;

最后闭环的话, 就是都要把新增的数据删除, 打开的界面都要关闭, 就比如原来我们做的自动化测试, 有一个申请借款的功能模块,

首先, 我们先进行自动化框架模块的分类划分, 分成需要定位的元素 page\_element, 公共模块 common(比如说数据库操作, 读取 excel 数据的操作, 都会封装成一个公共模块的)、测试数据 data、测试用例、元素操作 element\_action、执行用例以及报告文件等等这些模块。

先导入 selenium 包, 采用 css 定位, 把申请借款所对应的元素定位出来, 里面有一些内嵌界面, 需要先进入内嵌界面 --driver.switch\_to.frame(), 还有下拉框元素的定位操作(定位下拉框的话首先要导入 select 包, 通过 index, value, visible\_text 进行

选取值: Select(变量). Select\_by\_value('')把定位到的所有界面元素封装到 page\_element 模块里面, 方便后期维护。

对于申请借款的数据, 申请借款的一些标题, 利率等等, 我们都会保存在 excel 表中, 对于 excel 表格操作, 导入 xlrd 包把读取数据公共方法, 保存到公共模块 common 中, 方便以后从 excel 表中获取数据, 申请借款成功以后, 我们会添加一个断言, 获取界面 title 值, 去判断界面的提示, 当然也可以获取提示申请成功 text 内容去判断, 也会做数据库断言, 检查数据库数据, 导入数据库操作包 pymysql, 把数据操作, 封装到公共方法,

去调用(数据库操作步骤: 1 打开数据库连接, 2 建立游标, 3 执行游标(sql), 4 返回获取的数据)

申请借款模块需要调用到登录模块, 我们先把登录模块封装好, 放到元素操作 element\_action 模块中, 方便维护, 借款申请完成后, 我们会从数据库中删除相关数据实现闭环, 减低后续维护, 再关闭所有打开的界面。

我们运行自动化脚本的时候是用的 pytest 框架进行

导入 patent, 导入 yamail 包

定义一个主函数 if \_\_name\_\_=="\_\_main\_\_", pytest.main()所有的用例脚本, 定义生成对应的 HTML 报告到 report 目录下, 最后通过 yagma 定义发送邮件

定义发送邮箱的登录信息-定义邮件主题, 内容, 附件位置-发送邮件到测试人员邮箱中

当后续出现错误用例, 也会调用 pytest--if 来运行失败的测试用例

我觉得自动化中最难的在于

元素的定位、内嵌界面、时间插件, 这些需要一些 js 操作, 去除元素属性, 进行操作



自动化中发现问题：

在申请借款中，开发人员写了调试代码的弹出框，没有删除，导致我们自动化脚本跑不通；  
充值过程中，添加新的支付方式，开发修改了代码，导致原有充值失败

## 11.2 什么是自动化测试？

把一些简单，重复的工作，用自动化去实现，减少成本原来手工完成事情，通过自动化脚本去替代

## 11.3 什么样的项目合适做自动化(至少三个)

功能需求，需求稳定，界面稳定，不会频繁变动，才可以做

版本迭代次数比较多，做出来才有意义

项目周期比较长(大项目)意义

## 11.4 自动化测试流程是什么？

1、前期项目组会评估项目是否合适做自动化测试，例外就是看下哪些模块需要做自动化，适合做

2、人员安排，分工

3、确定测试框架

4、各自领取自己负责模块（根据功能来进行划分，功能测试的模块）

整理功能模块的用例，把能做自动化的用例提炼出，规整成自动化用例文档，

准备数据，整理数据，管理

资源，环境的准备

5、搭建测试环境

6、脚本开发

## 11.5 自动化实现的比例，1-2 年。

2 年以内，把原来手工做的事情 15-30%用自动化脚本去替代

2 年以上，实现 40-70%

## 11.6 自动化测试用例的来源

手工编写测试用例

把原来手工的测试用例，当成自动化测试用例

## 11.7 自动化测试的优点与缺点

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、提高测试效率和准确性，节约时间成本
- 4、可以执行一些手工测试困难或不可能进行的测试(例如，对于大量用户的测试，可以通过自动化测试模拟同时有许多用户，从而达到测试的目的)
- 5、自动化测试不存在执行过程中的疏忽和错误，从而增加软件信任度

**缺点：**

- 1、不能取代手工测试
- 2、手工测试比自动测试发现的缺陷更多
- 3、对测试质量的依赖性极大
- 4、由于自动测试比手动测试更脆弱，所以维护会受到制约，从而会制约软件的开发
- 5、工具本身无想像力

### 11.8 你们自动化覆盖率有多少？

2 年以内，把原来手工做的事情 30%用自动化脚本去替代

2 年以上，实现 40-70%

当时覆盖的确实也不是很多，差不多 30-40%的样式吧

### 11.9 自动化测试环境的搭建是怎样的？

- 1、首先要安装 Python
- 2、然后安装编写代码工具 pycharm
- 3、导入 selenium 库(`pip install selenium` 在 pycharm 安装 selenium3 下载源码包)
- 4、下载浏览器驱动(谷歌、火狐等)，把驱动程序发到 Python 根目录下

### 11.10 selenium 的原理是什么？

我们用的 selenium3x 以上的版本，对于 selenium2x 以上的版本原理是这样的：

Selenium2.0 则是把 selenium1.0 中 selenium RC 替换为了 WebDriver

WebDriver 利用浏览器原生的 API，封装成一套更加面向对象的 SeleniumWeb Driver API

直接操作浏览器页面里的元素，甚至操作浏览器本身(截屏，窗口大小，启动，关闭，安装插件，配置证书之类的)，由于使用的是浏览器原生的 API 速度大大提高，而且调用的稳定性交给了浏览器厂商本身，显然是更加科学，然而带来的一些副作用就是，不同的浏览器厂商，对 Web 元素的操作和呈现多少会有一些差异，这就直接导致了 SeleniumWebDriver 要分浏览器厂商不同，而提供不同的实现，例如 Firefox 就有专门的 FirefoxDriver，

Chrome 就有专门的 ChromeDriver 等等

### 11.11 Selenium2 与 Selenium1 的区别是什么?

Selenium1.0 使用的是 Javascript 注入技术与浏览器打交道, 需要 Selenium 启动一个 Server, 将操作 Web 元素的 AP 调用转化为一段段 Javascript, 在 Selenium 内核启动浏览器之后注入这段 Javascript, 开发过 Web 应用的人都知道, Javascript 可以获取并调用页面的任何元素, 自如的进行操作, 由此才实现了 Selenium 的目的: 自动化 Web 操作, 这种 Javascript 注入技术的缺点是速度不理想, 而且稳定性大大依赖于 Selenium 内核, 对 API 翻译成的 Javascript 质量高低 Selenium2.0 则是把 selenium1.0 中 selenium RC 替换为了 Web Driver

WebDriver 利用浏览器原生的 API, 封装成一套更加面向对象的 SeleniumWebDriverAPI 直接操作浏览器页面里的元素, 甚至操作浏览器本身(截屏, 窗口大小, 启动, 关闭, 安装插件, 配置证书之类的), 由于使用的是浏览器原生的 AP, 速度大大提高, 而且调用的稳定性交给了浏览器厂商本身, 显然是更加科学, 然而带来的一些副作用就是, 不同的浏览器厂商, 对 Web 元素的操作和呈现多少会有一些差异, 这就直接导致了 SeleniumWebDriver 要分浏览器厂商不同, 而提供不同的实现例 Firefox 就有专门的 Firefox Driver,

Chrome 就有专门的 ChromeDriver 等等

### 11.13 定位元素的 8 个方法是什么?

1、通过 id 定位

`find_element_by_id()`

2、通过 name 定位

`find_element_by_name()`

3、通过 class name 定位

`find_element_by_class_name()`

4、通过 css 定位

`find_element_by_css_selector()`

5、通过 link 定位

`find_element_by_link_text('文本')` 没有空格一般 a 链接

6、通过 partial\_link 定位

`find_element_by_partial_link_text(部分文本)` 一般有空格 a 链接

7、通过 xpath 定位

`find_element_by_xpath()`

8、通过 tag\_name 定位

`find_element_by_tag_name()`

### 11.28 定位不到元素，你碰到过哪些?怎么解决的?

1、像页面加载延迟的问题，这个需要通过等待延迟的方式来处理。

2、不过有时候，页面加载完成，但是元素暂时还不可见，导致定位不成功

这个可以选择使用显示等待来处理，这里需要用到 `WebDriverWait` 类来实现

3、还有就是像内嵌网页的问题，需要使用 `driver.switch_to.frame(name/index)` 这个函数来跳转到处理。

4、还有要注意多窗口问题，动态 id 问题等的问题，对于多窗口处理，可以使用

`driver.switch_to.window()` 的方式来进行处理，而对于动态 id 的问题，需要注意的是有些 id 跟数字有关，可能会动态变化，可以使用 `xpath` 也可以使用 `css_select` 属性定位或者样式定位，或者可以通过父元素来找元素，或者通过兄弟节点来找对应的元素。等等

5、还有要特别注意滚动条的问题，这里通过调用 js 代码来实现的，`driver.execute_script(js)`

6、再这就是有时候会碰到某些元素的是不可见的，比如 `display` 属性为 `none` 这就需要通过 `javaScript` 修改 `display` 的值。

```
js='document. querySelectorAll("select")[1]. style. display="block";'  
driver.execute_script(js)
```

### 11.29 元素定位，有时候定位得到，有时候定位不到，可能是什么原因，你会怎么处理?

1、可能是网络问题，导致页面加载延迟，这个可以做延迟等待，一般选择隐式等待，在脚本前面加上 `driver.implicitly_wait(20)`。

2、也有可能是页面结构发生变化导致的，这个时候最好选择通过 `xpath` 或 `css` 结合属性进行或者样式定位可能会好点，或者采用 `JQuery` 定位的方式来进行定位元素

### 11.30 对于不可见/隐藏的元素，你如何定位，如何处理?

我们可以通过 `javaScript` 修改 `display` 的值，来实现

#修改元素的属性

#我们需要通过 `javaScript` 修改 `display` 的值。

```
js='document. querySelectorAll ('select')[1]. style. display="block";'  
driver.execute_script(js)
```

### 11.31 如果元素经常变动你怎么做?

1，少用绝对路径，多用 `css` 或者 `jQuery` 定位

2，单个元素属性进行封装

### 11.32 自动化如何破解验证码

- 1, 开发去掉验证码
- 2, 开发设定一个万能验证码
- 3, python 导入 pillow 和 OCR 模块 pytesseract

### 11.33 数据驱动有没有了解过, 具体怎么做的?

- 1、其实就是把数据与脚本分离, 好处就是方便维护管理, 后期数据有变动, 只要改 excel 表的数据就可以, 脚本不用动
  - 2、封装一个读取 excel 表格的函数方法, 调用这个函数来读取数据。
  - 3、然后利用 ddt 模型, 使用 dt 内部装饰器来实现数据引用
- 我们当时就是这么做的。

### 11.34 参数化的类型有哪些?自动化脚本如何实现数据的参数化?

把测试数据, 通过变量或者文件进行保存, 通过修改文件或者变量, 达到数据参数化目的  
测试过程, 减低数据维护成本

- 1、像测试用例数据这块, 我们都是放在 excel 表格中的
  - 2、常用的变化比较小的配置数据, 比如数据库连接的配置, 报告的路径, 用例的路径等等
- 这些放在 config 配置文件中, 利用全局变量来管理的。对于全局变量的数据, 直接导入 config 某块, 调用就可以。目的是以便于测试数据改变时, 直接修改变量就行了

### 11.35 随机数如何操作?

1. 首先都要 import random

随机整数: random.randint(10002000), 有时需要转化成字符串类型:

```
str(random.randint(1000, 2000))
```

随机字符串: "join( random.sample( 'abcdefghijk' 6)), 有时前面固定字母了:

```
'ldh'+"".join(random.sample( 'abcdefghijk' ,6))
```

### 11.36 自动化中如何去操作 excel 表格?

需要用到 xlrd 库, 调用这个库中的 API 函数来实现的。

1. 第一步: 导包 import xlrd
2. 第二步: book= xrd.open\_workbook(文件路径)
3. 第三步: table= book.sheet\_by\_name 表名)
4. 第四步: 读某行数据一般都技行来读取

Table.row\_values(x)

如果要读全部的数据，多行数据，利用循环读取就可以

List=[]

for i in range(1,table.nrows):

List.append(table.row\_values(i))

return list

### 11.37 如何去读取表格中一行，一列数据，具体数据

1、读取一行数据：YS.row\_values(0) 0 表示第一行

2、读取一列数据：YS.col\_values(0) 0 表示第一行

读取具体某个数据：YS.row\_values(0)[0]--表示获取第一行第一列

### 11.38 如何读取表格中总的列数跟行数

获取总列数：YS.ncols

获取总行数：YS.nrows

### 11.39 自动化中如何去操作数据库

需要自己封装一个函数

1、安装 pymysql: pip install pymysql 或者在 pycharm 里面安装 pymysql

2、导入 mysql: import pymysql

3、连接数据库: db\_connect= pymysql.connect(host='ip 地址',  
user='root', password='', database='', port='', charset='utf8')

4、建立游标: cursor= db\_connect.cursor()

5、执行游标: cursor.execute(sql)

6、获取数据

获取游标中所有数据: all= cursor.fetchall()

获取游标中多条数据: many= cursor.Fetchmany()

获取游标中第一条所有数据: one= cursor.fenhone()

7、关闭游标: cursor.close()

8、关闭数据库连接 db\_connect.close()

### 11.40 如何获取数据库返回的所有，一行，前 5 行

获取游标中多条数据: many= cursor.fetchmany(5)

获取游标中第一条所有数据: `one= cursor.fetchone()`

首先已经建立好了数据库连接的函数, 并且 `return` 了

## 11.41 数据库返回的数据格式是什么类型的格式

二维元组的格式

如果要获取具体某个值要用: `cursor.fetchall[0][0]`---这个表示提取游标中所有值里面的第一个元祖里面的第一个值

## 11.42 ui 自动化你是怎么做断言的?

断言主要检查几个点:

1. 页面跳转是否正确(`title,url`)
2. 页面数据是否正确(核心数据, 页面上的关键信息, 或者页面上的某个元素是否出现)
3. 如果要检查数据库, 需要连接数据库, 查询数据, 与预期结果进行对比

基本都调用 `unittest` 框架中提供的断言函数来实现的, 用得比较多的就是

`AssertEqual`

`AssertIn`

#扩展

1. 提取元素 `text` 断言

`User_text= driver.find_element_by_css_selector("# j_account>a").text`#获取实际的值(展示账户名内容)

`assert 'admin' in user_text` #断言 `admin` 在实际值里

2. 提取元素属性去断言:

`login_button_value=driver.find_element_by_id("lajax-login-submit").get_attribute("vaue")`

#获取登录按钮的属性 `vaue` 值是登录

`assert '登录' == login_button_value` #断言登录是这个按钮的属性

3. 提取界面 `title` 值去断言

`assert "p2p 信贷一最大、最安全的网络借贷平台" in driver.title`, "失败"或者 `in` 改为 `not in`

4. 提取元素是否可用来进行断言

`Assert driver.find_element_by_css_selector("div.user_money > a : nth-child(1)").is_enabled0==True`, "元素不存在. 不可用"

5. 数据库断言, 去数据库查询结果, 是否跟预期一致

### 11.43 断言有哪几种(三种)，常见 assert 断言

selenium 提供了三种模式的断言：assert、verify、waitfor

Assert 失败时，该测试将终止。

Verify 失败时，该测试将继续执行，并将错误记入日显示屏。也就是说允许此单个验证通过。确保应用程序在正确的页面上。

Waitfor 用于等待某些条件变为真。可用于 AJAX 应用程序的测试

常见的 assert 断言：

assertTitle(检查当前页面的 title 是否正确)

assertText(检查指定元素的文本)

assert Attribute(检查当前指定元素的属性的值)

如果该条件为真，他们将立即成功执行。如果该条件不为真，则将失败并暂停测试。直到超过当前所设定的超过时间。一般跟 setTimeout 时间一起使用

## 十二、Unittest 框架

### 12.1 你们自动化用例是怎么管理的？

1. 所有用例都是放在 test\_case 的目录下的统一管理的。
2. 每个某块一个.py 文件，数据独立放在 excel 表格中
3. 所有的具体用例脚本都是依据 unittest 来编写的，利用 ddt 模型的装饰器来引用数据
4. 然后跑用例这块，主要用的是 unittest 框架来统一加载用例，并执行用例的。

如果要全量跑，调用 unittest 中的 defaultTestLoader.discover 这个函数来加载 test\_case 目录下的所有.py 文件。

### 12.2 Web UI 自动化都用到过哪些库？

Selenium 库 里面封装了丰富的对浏览器，页面元素进行操作的方法。

Xlrd 库 主要用来实现对 excel 表格数据进行读取的 API

Pymysql 库 主要用来操作数据库的

Ddt 库 主要用来实现数据驱动的

Re 库 主要用来提取 html 页面数据的

Unittest 库 主要用来编写用例，管理用例，执行用例的。

### 12.3 Unittest 框架的原理？

Unittest 框架有几个大组件，1. 测试固件 ( setUp, tearDown)



2. 测试用例 3. 测试套件 4. 加载器 5. 运行器 6. 测试结果

首先我们需要创建测试用例，然后利用加载器讲用例加载到测试套件中，并创建一个执行器，去执行测试条件中的所有用例。

它可以帮我们进行管理用例，统计加载执行用例，批量跑用例。

## 12.4 unittest 框架有哪些组件？

**test fixture(测试固件):**

包含一个 `Setup()` 方法/函数，`tearDown()` 方法/函数，用例执行之前都会先执行 `Setup()` 方法/函数主要是完成一些准备初始化的工作，比如创建临时的数据库，文件和目录，用例数据读取，浏览器的打开等，用例执行完成之后，会执行 `tearDown()` 方法/函数，完成一些清理回收的工作，比如数据库断开，关闭浏览器。

(1) 比如说在这个测试用例中需要访问数据库，那么可以在 `setUp()` 中建立数据库连接以及进行一些初始化，在 `tearDown()` 中清除在数据库中产生的数据，然后关闭连接，注意 `tearDown` 的过程很重要，要为以后的 `TestCase` 留下一个干净的环境。

**test case(测试用例):**

什么是测试用例呢？

就是一个完整的测试流程包括测试前准备环境的搭建(`setUp`)，以及测试后环境的还原(`tearDown`)，还有包括用例方法，每个用例方法都必须要以 `test` 开头。

**test suite(测试套件):**

多个测试用例的集合就是 `suite`，一个 `suite` 可以包含多个测试用例，也可以嵌套 `suite`。可以通过 `addTest()` 方法手动增加 `TestCase`，也可通过 `TestLoader` 自动添加 `TestCase`，`TestLoader` 在添加用例时，会没有顺序。

**test runner(运行器):**

用来执行测试套件中测试用例的，最终执行完成之后会生成一个测试结果。

`TestLoader`(加载器): 用来加载用例，把用例加载到测试套件中

`Test Result`(测试结果): 包括运行了多少测试用例，成功了多少，失败了多少等信息。

## 12.5 unittest 框架如何使用？

### 1. 导包

```
import unittest
from selenium import webdriver
import ddt
```

2. 定义一个类继承 `unittest.TestCase` 基类

3. 重写 `setUp()`, `tearDown()` 方法

`setUp()` 方法实现一个初始化的准备工作，比如，实例化 `webdriver` 对象，对 `driver` 进行初始化配置，连接数据库.....

`tearDown()` 方法实现释放资源的任务。

4. 编写用例方法，用例方法必须以 `test` 开头

5. `Unittest` 如何去运行多个文件或者整个目录

因为我们用例全部是放在 `test_case` 目录下统一管理的，基本每个某块都是一个 `.py` 文件，要全量跑的话，需要调用 `unittest.default.discover()` 函数，指定用例目录的路径，加载所有的 `.py` 文件，它会自动创建测试套件，并把用例加入测试套件中，然后利用 `unittest.TestRunner()` 创建一个执行器利用这个执行器去运行测试套件中的所有用例。

## 12.6 如何生成自动化测试报告？

我们当时用的是 `HtmlReport` 这个库来生成自动化测试报告的。

1. 安装

```
pip install HTMLReport
```

2. 使用方法

# 测试用例执行器

```
runner= HTMLReport.TestRunner(  
    Report_file_name='test',    #报告文件名，如果未赋值，将采用“test+时间戳”  
    Output_path='report',      #保存文件夹名，默认“report”  
    title=' 测试报告' ,        #报告标题，默认“测试报告”  
    description=' 无测试描述'   #报告描述，默认“测试描述”  
    Thread_count=1,            #并发线程数量(无序执行测试)，默认数量 1  
    Thread_start_wait=3,       #各线程启动延迟，默认 0s  
    Sequential_execution=False, #是否按照套件添加( addTests)顺序执行，  
    #会等待一个 addTests 执行完成，再执行下一个，默认 False  
    #如果用例中存在 tearDownClass，建议设置为 True，  
    #否则 tearDownClass 将会在所有用例线程执行完后才会执行  
    # lang='e  
    lang='cn' #支持中文与英文，默认中文  
)
```

```
#执行测试用例套件
```

```
runner.run(suite)
```

## 十四、性能测试

### 14.1 性能测试怎么测试

性能测试其实就是通过自动化工具模拟多种正常、峰值以及异常负载来对系统的各项性能指标进行测试。负载测试和压力测试都属于性能测试，二者可结合使用。

性能指标主要有平均响应时间、90%响应时间、吞吐量、吞吐率，每秒事务数，以及服务器的资源使用率(CPU 占比，mem 内存占比等)等。当并发用户数超过 300 时，为了让测试数据更准确，可以考虑分布式压测，通过司令机控制几台奴隶机进行测试。

性能测试要先调试好脚本，主要考虑对脚本的数据参数化和添加断言。因为有些接口需要对业务逻辑或参数格式进行校验，为了能让所有线程数跑起来，需要将数据参数化。

数据参数化有这几种做法：

- 1、可以将一些固定值改成随机函数；
- 2、利用 JDBC 从数据库读取数据并关联变量；
- 3、Excel 数据参数化，
- 4、动态关联参数化，断言是为了判断用例是否

执行成功，并验证服务器的响应错误率。响应断言常用 json 断言，xml 断言用的最少，

性能测试的目的是为了检验系统能否满足客户的性能需求，若性能需求无法满足时，则要考虑对系统进行性能调优，一般用排除法：

- 1、首先考虑网络方面问题：使用 ping 命令查看与目标服务器的连接是否正常、传输速度的快慢。通过提升服务器的带宽，看响应时间是否相应降低。
- 2、考虑数据库的问题，可以单独去压测数据库，查看数据库的最大连接数和 SQL 语句的执行时间，索引命中率和 sleep 等待时间等
- 3、考虑 Apache 中间件的问题，查看中间件设置的最大连接数是否合理，如果设置的连接数太小，会话数超过设定的最大连接数时会导致等待时间变长，出现响应超时情况
- 4、考虑服务器的硬件配置，如内存、CPU、mem、磁盘读写速度等，可以用 top 命令来监控，也可以使用 nmom 工具来监控，nmom 会把监控的数据形成表格形式，方便我们查看。
- 5、最后考虑开发代码写的好不好，处理时间长不长的的问题。

举例：在我之前的公司，我们主要是会考虑用户操作使用比较频繁的模块，比如借贷，充值，投资模块，我们一般会通过增加并发数来压测，观察 CPU、mem、磁盘读写、吞吐量和每秒事务数等性能指标，以前我老大要求我并发 100 个用户，我用 jmeter 把线程数设为 100，永久循环，持续时间半个

小时，设置启动延迟 55，在 Linux 启用 nmom 工具监控服务器。

当我运行脚本的时候我看聚合报告 90% 的平均响应时间达到了 6s，吞吐量也比较小，用 top 命令监控资源发现 CPU 差不多到了 100%。于是我用 Navicat 工具通过 SQL 命令 show full processlist 取当前运行的 SQL 语句，发现有许多语句用的是左关联，在查看了这条 SQL 语句的执行计划发现没有用索引，再查看了索引的命中率，命中率倒是还行看了下 nmom 生成的报告，发现 CPU 一直是处于爆满状态，其中主要是 mysql 的占比很大，这个时候我基本上判断数据库的问题。

于是我就照着前面的步骤再次压测，同样还是用 nmom 工具去监控 CPU，mem 网络等状态，这次我主要是在 Navicat 上用命令去抓取 SQL 语句，还是一样有很多语句都是左关联，并发现很多空连接 (sleep)，我就用 show global variable like "wait\_time" 命令查看了设置的休眠时间 (等待时间) 发现时间很长 28800s，然后我就把这个休眠时间改成了 20s，因为 SQL 语句使用了很多左连接，我就用 show variables like "tables\_size" 查看了临时表的空间大小、发现临时表只有 16m，我将空间改成了 1G 再去压测了下，发现 CPU 只是降了 10% 左右，nmom 报告上还是显示 mysql 占的 CPU 很大，然后运行的时候，用 top 命令监控，发现有的时候有很多 mysql 进程同时运行 (因为没有设置连接池)，我就用命令查看了下 mysql 的最大连接数，因为 SQL 语句的执行速度还是挺快的，所以就把 mysql 的连接数调小到 50，再去跑了一遍发现 CPU 降到了 40% 左右，并且其他的性能指标也都还不错。最后把聚合报告的数据以及 nmom 的数据整理成性能报告给老大，其实做接口性能主要就是用排除法一个一个去排除，发现性能问题就要先解决了性能问题再压测，不然其他的问题也有可能是这个性能问题导致的所以接口性能基本上就是观察，各个性能指标都在范围之内就差不多了。

## 14.2 性能测试流程是怎么样的？

例外一种问法：简单介绍下你们公司的性能测试流程是怎么样的？

我们那个项目的性能做得不多，公司要求也不严格。

对于流程这块，首先就要对整个系统进行详细的分析，确定基本的测试范围，看下系统的哪些业务是需要做性能测试的，还有就是做那方面的性能测试，对于我们那个项目，当时就做了几个业务做了些简单的并发压测 (稳定性) 这块，像登录的，搜索查询，下单，还有就是购物车里面的几个接口都有做过，然后就是对各个业务场景进行详细的场景分析与设计，确定每个业务场景的并发数，是否需要设置集合点啊，压测时间是多长，还有各个业务场景的性能指标等等，场景设计这块基本上都是老大跟产品哪个一起弄的，我参与的不是太多。

上面把个场景设置好了之后，提交给我们，我们就是根据老大设置好的那些场景编写了基本的性能测试用例，其实做性能测试，我觉得前期最关键的还是业务场景一定要设计好，后期我们主要的任务就是准备各自任务需要用到的一些测试数据，搭建好测试环境，还有就是测试脚本设计与开发，执行，并生出测试报告，对于测试结果我们一般会简单的做个分析，如果没有什么问题，基本后期就写一个

性能测试报告。流程大概就是这样的。

### 14.3 你们性能观察哪些指标，大概指标范围是怎么样的。

对于指标这块，业务方面的指标主要有：并发数，90%用户的平均响应时间

错误率，吞吐量/吞吐率这些，例外还需要关注服务器资源的使用情况，像：CPU 的使用率、内存的占有率，磁盘 IO，网络。

我们那个项目当时只针对，登录，搜索查询，下订单，购物车相关接口，支付等业务做了些简单的并发，压测这块，指标大概是这样的：

单基准业务并发测试登录，注册，查询 1s 以内，下订单，购物车相关接口，支付 2s 以内，混合业务性能：5s 以内

响应时间：登录，注册业务<2s 之内查询，下订单，购物车，支付业务<3s

充值，提现，查看充值日志，查看提现日志业务查询标的，<3s

投标，申请借款<5s

错误率：0

吞吐量/吞吐率：200 左右请求/sec

CPU：80%以内

内存：80%以内

I/O： %util<=80%，%nowait<=20%

%util：磁盘一秒中有百分之多少的时间用于 I/O 操作，

% nowait：磁盘等待处理时间占比

带宽：<=系统带宽的 30%，无丢包，无延迟，无阻塞

### 14.4 这个测试的环境配置，如转速度

租用的服务器，一台数据库服务器，一台后端服务器

8 核 16G 网络带宽 100M，2.5GHZ 磁盘 15000pm 转数

### 14.5 性能测试计划有哪些内容

写过，主要是时间进度安排与工作安排，主要是环境，测试任务，测试需求，测试方法与策略，测试环境准备，测试通过的标准。

比如说原来我们一个项目性能测试时做了 5 天，那我们计划是，测试策略与用例编写一天，测试准备需要 1 天，测试执行 2 天，报告总结 1 天。

## 14.6 有没有写过性能测试报告，具体包括哪些内容

性能测试报告，需要每次 Jmeter 压测完成的 html 报告的数据跟 nmon 工具监控的数据，整理出一份性能测试报告，性能测试报告，主要包含：

- 1, 测试资源(环境，测试数据，表里面需要多少数据，测试工具)
- 2, 测试设计(测试业务，测试类型，测试时间，并发用户数)
- 3, 测试分析(每一个场景都需要分析)
- 4, 测试结论(能不能上线，不上线的原因)
- 5, 优化和建议
- 6, 测试通过的标准，平均响应时间<5s，资源利用率<75%，事务失败率<5%

## 14.7 什么是内存泄漏，什么是内存溢出？

内存泄漏：

是指程序在申请内存后，无法释放已申请的内存空间，导致系统无法及时回收内存并且分配给其他进程使用。通常少次数的内存无法及时回收并不会到程序造成什么影响，但是如果在内存本身就比较少获取多次导致内存无法正常回收时，就会导致内存不够用，最终导致内存溢出。

内存溢出：OOM

1. 指程序申请内存时，没有足够的内存供申请者使用 1M 实际要占用 2M 内存，就说分配的内存不够，导致内存溢出
2. 给了你一块存储 int 类型数据的存储空间，但是你却存储 long 类型的数据
3. 长期出现内存泄漏，导致系统内存越用越少，最终导致内存不够用，导致系统崩溃，出现 OOM

## 14.8 吞吐量，吞吐率

吞吐量：KB

指在一次性能测试过程中网络上传输的数据量的总和(单位应该 KB)也可以这样说，在单次业务中，客户端与服务器端进行的数据交互总量；对交互式应用来说，吞吐量指标反映服务器承受的压力。

并不是吞吐量越高越高，一个服务器的性能，要从多个方面去考虑：

90%用户的平均响应时间、错误率、吞吐量/吞吐率、CPU、内存、磁盘 I/O、网络的占用情况，还有服务器的配置。

吞吐率：

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

12s 800M 数

$800 * 1024 / 12 = 66666 \text{ KB/sec}$

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

#### 14.9 吞吐量与吞吐率跟负载有什么关系？

吞吐量/率和负载之间的关系：

- 1、上升阶段：吞吐量随着负载的增加而增加，吞吐量和负载成正比；
- 2、平稳阶段：吞吐量随着负载的增加而保持稳定，无太大变化或波动；
- 3、下降阶段：吞吐量随着负载的增加而下降，吞吐量和负载成反比；

总结：吞吐量/率干不过负载!!!

#### 14.10 当你服务器满了之后，你们吞吐量和响应时间怎么变化的

吞吐量会所有下降，响应时间会变长

#### 14.11 你们的 TPS 的指标是什么估算的？

假设一个系统的业务有登录、浏览帖子、发送新帖、回复帖子，访问高峰是上午 10 点，日访问高峰 PV 约 5208(含登录 1300、浏览 2706、发帖 526、回帖 676)，系统响应时间要求小于 3 秒，试计算此系统的 TPS 以及并发数。

如果分析业务量的数据是以 PV 来统计的，我们需要把 PV 转化为 TPS。

PV 的统计一般可以通过监控埋点或者统计访问日志统计得出。

说到 PV 还有个特殊的情况，叫 PeakY，指一天中 PV 数达到的高峰 PV 值。

通过一些监控系统，也可以直观看到统计数据。

实际上一个 PV 即一次对服务器的客户请求，可能还包含了很多资源请求，比如图片、样式 JS 信息、文字等。而浏览器具有资源缓存功能，下次访问同样资源将不会再从远程服务器上下载，这大大加快了响应速度。如果我们使用代理服务器访问外网资源时，多数代理服务器也会缓存这些静态数据。也就是说浏览器与服务器之间的动态数据的 Size 会小于静态数据。

所以浏览器是否缓存了静态数据对性能测试影响明显。我们在做性能测试时，其中就有许多用户可能是新用户，在他们的浏览器上还没有缓存这些静态数据，为了更准确的模拟用户请求，我们有必要不缓存这些静态内容。所以性能测试中是否缓存访问的静态资源要根据业务情况而定。

本例中，我们把一个请求放在一个事务中来统计服务器的响应时间。这么说，一个 PV 即是一个事务(每秒的 PV 量并不直接等同于 TPS，因为一次客户请求可能包含了很多资源请求。如果我们不关心页面刷新时请求资源的耗时，此时我们就把每秒 PV 数等同于 TPS)，比如一个功能页面(浏览帖子)每秒会有 10 个 PV，那么此功能的 TPS 即为 10。

### 估算 TPS:

业务量一般要取系统业务高峰的值，才能代表系统的实际处理能力，系统在 10 点的访问高峰 PV 约 5208，那么这个时段的  $TPS = 5208/3600 \approx 1.45$  吗？

答案是否定的，因为在一小时内的吞吐量未必是平均的。

如果我们采集到的业务量数据能够细分到每分钟，TPS 就越准确。如果不能，可以按照二八原则。即 80% 的业务在 20% 的时间内完成， $TPS = (5208 * 80%) / (3600 * 20%) \approx 5.8$

### 估算并发数:

1、由 TPS 进行估算      2、由在线活动用户数估算      3、根据经验估算

方法 1: 由 TPS 进行估算

因为  $TPS = \text{事务数} / \text{时间}$ ，假设所有的事务都来自不同的用户，那么并发数 = 事务数 =  $TPS * \text{时间}$ 。

具体如下:

$Vu(\text{业务名称}) = TPS(\text{业务名称}) * (Runtime + ThinkTime)$

其中， $Vu(\text{业务名称})$  表示此业务的虚拟用户数，即并发数。RunTime 是测试程序/脚本运行一次所消耗的时间，包括事务时间+非事务时间。ThinkTime 是模拟用户思考或者填写表单消耗的时间。

下面是发帖动作的测试脚本伪代码(T、TT、AT 表示时间，单位为秒，AT 一般都是非常小的，包含在事务时间中，可以忽略).  $Runtime = T1 + \dots + T7$ ;  $ThinkTime = TT1 + TT2$ 。

Login

Enter login page (进入登录页面)  $T1 = 0.2$

ThinkTime (思考时间)  $TT1 = 3$

Declare login transaction (声明登录事务)  $T2 = 3$

Commit form (提交登录表单)

Assert login (检查登录是否成功)  $AT1$

Enter default page (进入登录成功后的默认页)  $T3 = 0.2$

Sender topic

Enter topic list (进入论坛列表)  $T4 = 0.2$

Enter new page for topic (选入新帖编辑页面)  $T5 = 0.2$

ThinkTime (思考时间)  $TT2 = 3$

Declare newTopic transaction (声明新帖事务)  $T6 = 3$

Commit fom (提交新帖)

Assert commit (查发帖是否成功)  $AT2$

Enter topic list (提交新帖后进入论坛列表)  $T7 = 0.2$



业内一般把 Think Time 设为 3 秒。测试需求中要求响应时间小于 3 秒，我们就以 3 秒为阈值

可得  $V_u = \text{TPS} (\text{RunTime} + \text{ThinkTime}) = 5.8 * (T_1 + TT_1 + T_2 + T_3 + T_4 + T_5 + TT_2 + T_6 + T_7) \approx 76$

如果只计算事务时间， $V_u = \text{TPS} * (T_2 + T_3) = 34.8 \approx 35$

可以看到两者之间的  $V_u$  数量相差巨大。由于一次迭代的时间会大于事务的响应时间，如果在估算时不把非事务消耗的时间加入进去，计算出来的 12 个并发用户在测试执行时很有可能无法达到  $\text{TPS} = 5.8$  的目标。

由于我们计算  $V_u$  时，使用的是系统 TPS (登录、浏览、发帖、回帖合计)，其中又以发帖业务消耗时间最长，所以计算出来的 76 个并发数实际上是系统业务的总并发数，需要按比例分配到各个业务。

| 业务名称 | 高峰业务量 | TPS  | 并发数 | 响应时间 | 事务成功率 |
|------|-------|------|-----|------|-------|
| 登录   | 1300  | 1.44 | 20  | <3 秒 | >99%  |
| 浏览   | 2706  | 3.0  | 40  | <3 秒 | >99%  |
| 发帖   | 526   | 0.58 | 7   | <3 秒 | >99%  |
| 回帖   | 676   | 0.75 | 10  | <3 秒 | >99%  |
| 合计   | 5208  | 5.8  | 77  | -    | -     |

#### 方法 2：在线活动用户数估算

1. 在线用户数的预估可以采取 20% 的系统用户数。例如某个系统的系统用户数有 1000，则同时在线用户数据有可能达到 200，或者预估 200 做参考。

在线用户数和并发用户数又存在着关系，即：平均并发用户数为  $C = NLTL$  为在线时长， $T$  为考核时长  
例如考核时长为 1 天，即 8 小时，但是用户平均在线时长为 2 小时，则  $c = n * 2 / 8$

$n$  为登录系统的用户数， $L$  为登录的时常。例如一个系统有 400 个用户登录，然后每个用户登录大概停留 2 小时，则以一天 8 小时考核，算平均并发用户则为  $C = 400 * 2 / 8$

答：我们系统的 TPS 当时是根据 PV 值来计算的，就是通过查看系统访问日志来获取每个业务功能每天高峰期的 PV 值 (也就是每天高峰期的时间段，有多少服务器的客户请求)，当时的 PV 值大概在 5000 多

然后按照二八原则，即 80% 的业务在 20% 的时间内完成， $\text{TPS} = (P_{80\%}) / (\text{时间 } 20\%) \approx 5.6$ 。算出来大概在 5.6。

[具体的指标都是 SE 跟测试主管他们经过分析出来给到我们的。他们开会的时候大概跟我们说过估算方式，差不多是这样的……]

#### 14.13 每人说一个项目接口，你设置多少并发

首先涉及到并发用户数可以从以下几个方面去做数据判断

1. 系统用户数(注册用户数量)
2. 在线用户数(平均每天大概有多少个用户要访问该系统, 可以从系统日志中获得)
3. 并发用户数(高峰期的时候的在线用户数量)

### 三者之间的关系

1. 在线用户数的预估可以采取 20% 的系统用户数。例如某个系统在系统用户数有 1000, 则同时在线用户数据有可能达到 200, 或者预估 200 做参考
2. 在线用户数和并发用户数又存在着关系。即: 平均并发用户数为  $C=NUTL$  为在线时长,  $T$  为考核时长例如考核时长为 1 天, 即 8 小时, 但是用户平均在线时长为 2 小时, 则  $c=n2/8$   
 $n$  为登录系统的用户数,  $L$  为登录的时常, 例如一个系统有 400 个用户登录, 然后每个用户登录大概停留 2 小时, 则以一天 8 小时考核, 算平均并发用户则为  $c=400*2/8$

答: 就拿登录接口来讲吧

我们的并发数是根据注册用户数量及每天在线用户数综合来估算的, 我们系统当时注册用户数量大概是在 70 多万的样子, 不过这里其实有一些僵尸用户, 真正的用户大概在 60% 的样子, 也就是差不多, 46 万多一点, 通过查看系统访问日志, 高峰期的时候每天在线数, 用户数量大概差不多在 52000 多点吧, 按 52000 估算, 每个用户停留时间大概在 20 分钟左右, 大概平均每天同时在线用户数量在 2145 多。其中登录业务的占比大概在 10% 的样子, 同时在登录的大概 80% 的比例计算, 登录接口大概设置的并发数在 172 多的样子, 查询业务的占比大概在 30%, 查询接口大概设置的并发数在 510 的样子, 下单业务大概占比在 20%, 下单接口的并发数设定在 345 的样子。

[具体的指标都是 SE 跟测试主管他们经过分析出来给到我们的。他们开会的时候大概跟我们说过估算方式, 差不多是这样的...]

#### 14.14 你们吞吐量是多少, 响应时间是多少, 你设置了多少并发?

登录: 吞吐量大概在 13.5/sec 响应时间<1s, 设置的并发数 180 个并发数。

查询: 吞吐量大概在 36/sec 响应时间<3s, 设置的并发数 500 个并发数。

下单: 吞吐量大概在 25.6/sec 响应时间<3s, 设置的并发数 350 个并发。

#### 14.15 做并发你们一般 cpu 和内存是多少?

cpu 大概在 60% 多点, 内存大概占比在 65% 的样子。

#### 14.16 有没有做过稳定性测试

部分接口有做过稳定性测试。具体怎么做的?

稳定性测试主要就是看某个业务在高并发情况下是否能持续稳定运行嘛, 当时大部分的核心业务都有做过稳定性的, 这个需是把并发数设置为峰值, 然后循环次数设置为永远, 例外要开启调度器, 调度

器中的持续时间设定为 3600 秒，让它持续压测 1 个小时。看下接口的各项性能指标的变化，是否在预期的指标范围之内。

#### 14.17 5000 个人抢购，只能 50 个人能抢到，你怎么设计并发数的

并发数，按群内最大人数计算，利用二八原则， $5000 * 80\% = 4000$ ，并发数的峰值为 4000

#### 14.18 微信群里面发送红包，5000 个人群，只能 3000 个人能抢到，你怎么设计并发数的峰值

并发数，按群内最大人数计算，利用二八原则， $5000 * 80\% = 4000$ ，并发数的峰值为 4000

#### 14.19 20 并发 40 次循环怎么做？

线程数设置 20 个，循环次数 40

#### 14.20 我想从 200 慢慢加载到 300，到 400 怎么做

这个需要用到自定义线程组，自定义线程组最大的好处就在于做压测的时候，可以设置一些复杂的业务场景，具体设置的话，就是.....

#### 14.21 需要插入 500 条数据，你怎么插入

1、使用存储过程来实现

2、可以通过 JMeter 来实现，调用注册接口，线程数设置为 500，账号，密码可以通过 JMeter 中的随机函数 `randomString()`，`random()` 函数结合计数器来实现。

#### 14.22 响应超时，你是怎么定位的

这里一般我会采用排查法，首先考虑软件优化问题，之后考虑硬件问题，思路大概是这样的

1、首先考虑中间件(tomcat, Apache 的连接数的问题)，可以尝试增加连接数，看是否变化。

2、例外还有就是数据库的连接数问题，也可以尝试增加，看是否有变化。

3、要不就是看数据库的访问效率问题，这里要考虑数据库的操作是否添加索引，如果没有添加索引，可以让开发优化下数据库的访问速度，添加索引，或者优化 sql 语句。

4、再一个就可以尝试考虑后台代码的架构设计是否合理，代码算法是否足够优化。

5、如果以上问题都不能解决，那么只能考虑增加服务器的 CPU 内存，或者增加网络带宽，看响应时间是否可以得到优化。

大概的思路差不多就是这样的吧。

#### 14.23 压测返回数据报错，你怎么去定位的

1、如果是所有请求的数据报错，那肯定是脚本问题，认真检查脚本参数。

2、如果只是部分请求报错，那估计是性能问题了。

#### 14.24 你理解的性能调优是什么？

响应时间过长(排除法) -用户体验不好

1, 网络情况(提升带宽)

2, 服务器资源情况

3, 中间件类型

4, 中间件的连接数

5, 代码质量问题

6, 数据库类型 mysql--Redis

7, 数据库连接数

8, sql 语句执行时间

`show full PROCESSLIST` 查看哪些 sql 语句运行比慢

1、看表是否建立索引

2、运行 sql 语句是否需要 sq 优化

9, 索引命中率

低于 0.1%比较好，低于 0.01%分配比较多，适当减少缓存空间大小

`show GLOBAL STATUS like "key——read%"`

`Key_reads=568`

`Key_read_requests=1329114`

`Key_cache_miss_rate=key_reads/key_read_requests=0.000427=0.04%`

`show VARIABLES like "key_buffer_size"` #查看缓存空间大小

`set GLOBAL key_buffer_size=8000000` #设置缓存空间大小

10, sleep 过多

`show GLOBAL VARIABLES like "wait timeout%"`

`show GLOBAL VARIABLES like "interactive timeout%"`

`set GLOBAL interactive_timeout= 30`

`set GLOBAL wait_timeout=30`

11, 临时表空间大小

性能排优一些方面：

1, 排除法

2, 很多问题，都是同一个问题导致，需要一个个去排除优化，解决

资源使用率不足：（系统薄弱的位置）

- 1, 用户量较少
- 2, 网络问题
- 3, 中间件连接数不够
- 4, 代码质量问题
- 5, 数据库问题

资源使用率过高：（系统会崩溃）

- 1, 用户量比较大
- 2, 中间件连接数太大
- 3, 代码质量问题
- 4, 数据库问题

#### 14.25 如果要做万并发，你怎么做

那我们就需要考虑分布式压测，那需要准备几台测试机，

master 机器要设置。。。。

奴隶机要设置。。。。。

也可以租用云测平台进行测试

#### 14.26 如果用户并发要慢慢加载，你怎么设置的

设置并发数的时候，我们会设置启动时间，比如说设置 50 个并发用户数就是 50 个线程组，启动时间会设置成 10 秒，让用户慢慢启动起来

#### 14.27 并发用户数跟响应时间与吞吐的关系

- 1, 并发用户数越多，响应时间越长
- 2, 并发用户数越多，吞吐量会一直，增加，增加到一个临界点(系统瓶颈后)，不再增加，有少许的回落

### 十五、app 测试

#### 15.1 app 测试你具体怎么做的？

对于 App 这块，我们一般首先都先做功能，先保证功能过关是第一位，对于功能这块的话，基本都跟 Web 端是一样的。

除了功能之外，公司还要求做了一些专项测试，像：安装，卸载测试，兼容性测试，稳定性测试，性能测试，弱网测试，交互性测试，都有测试过的，专项测试这块，我主要负责的是：兼容性测试，稳

定性测试，性能测试，弱网测试，交互性测试，这是我这边负责的。

像兼容性测试，公司有提供了差不多了 7-8 款的真机，像：华为，小米，三星， vivo， oppo 等这些主流的机型都在真机想有测试过，其他的机型，公司用的是云测，云测平台我们用的 Testln 这个平台，公司会给我们提供账号。

稳定性测试这块，用的 Monkey 命令工具去测的，主要就是通过 monkey 模拟用户发送一些伪随机时间，看 app 是否有 Crash， ANR， Exception 等现象，一般都是在晚上的时候去执行 monkey 命令，然后出报告，分析性能测试，用的 GT 工具结合 Android Studio 工具去检测 app 在手机上运行的时候的 CPU，内存，电量，流量，启动时间，安装，卸载时间以及页面的响应时间。

弱网我们用的 fiddler 工具去进行模拟的，模拟 2G/3G/4G 等弱网场景，看 app 在弱网情况，功能是否能正常使用。

交互性测试这块主要就看 app 与其他应用程序之间的交互运行，以及与系统应用程序之间交互运行，来回进行前后台切换，看是否会出现闪退，数据丢失等现象。

## 15.2 Web 测试与 app 测试区别？

其实功能这块，app 测试与 Web 测试基本是一样，没有什么区别。(需求分析->提炼测试点>编写测试用例->执行用例->提 Bug->复测，回归)等等的。

区别主要在于，web 端是 B/S 架构的，App 是 C/S 架构的，由于架构的不同，所以 web 端一般服务器更新的时候，客户端不需要更新，

因为它通过浏览器来访问的，服务器更新了，客户端也更新。app 服务端要更新，同时客户端软件要进行升级更新，才算是新的版本。

对于 app 测试来讲，除了功能之外，更多的还要考虑一些专项测试，比如：

Web 测试是基于浏览器的所以不必考虑安装卸载。而 app 是客户端的，则必须测试安装、更新、卸载兼容性、稳定性、性能测试、弱网测试、交互性测试等等。

还有就是，对于兼容性这块，Web 端主要考虑是：不同的浏览器，不同的操作系统的兼容性接口。

而对于 app 测兼容性更多的考虑：不同的品牌机型，不同操作系统，不同手机屏幕大小，屏幕分辨率性能方面也会有所不同：Web 端性能测试更多关注的后台的性能，app 的性能测试关注的是手机本身的资源的性能问题：

比如：CPU 内存，电量，流量，页面加载响应时间，软件启动时间等等

他们两个之间的区别差不多就这些吧。

## 15.3 常用的 adb 的命令？

adb start-server

adb kill-server  
adb devices  
adb -s 设备 ID install 路径/包名.apk  
adb -s 设备 ID shell pm list packages -3  
adb -s 设备 ID uninstall com.baidu.BaiduMap  
#电脑端文件传输到手机上  
adb -s 设备 ID push D:\路径文件\ sdcard\路径\  
#手机上的文件传输到电脑端  
adb -s 设备 ID pull \sdcard\路径\文件\D:\路径  
#查看手机端的日志  
adb logcat  
adb logcat -d #打印完所有的日志文件之后，退出 shell 终端  
adb logcat -c #清除手机系统运行生成的日志文件  
adb logcat -v time #需要打印日志详细时间的简单数据  
adb logcat -d \*:E #需要打印级别为 Error 的信息  
adb logcat -d \*:E>D:\hello.log  
adb logcat -d \*:l>D:\hello555.log #打印 l 以上级别的所有日志信息  
adb logcat -d \*:E | findstr cn.csdn.activity > D:/hello\_error2.log  
查看所有手机软件包名 adb shell pm list package  
查看第三方的手机软件包名 adb shell pm list package -3  
查看后台运行的包名 adb shell am monitor  
查看手机当前使用的内存情况，各个线程的内存占用情况 adb shell dumpsys meminfo  
查看手机的电池信息 adb shell dumpsys batteryinfo  
查看系统资源状态 adb shell top  
adb 命令录屏：  
adb shell screenrecord --time-limit 10/sdcard/demo.mp4 (10 表示录制 10 秒，默认是 180 秒)

#### 15.4 adb 的作用的？

adb 其实是一个 android 调试桥，主要是用来监控手机设备的，实现手机端与电脑端的通信，通过 adb 来实现对手机的管控。比如：通过 adb 安装软件卸载软件，通过 adb 可以查看手机的资源使用情况，可以查看 cpu 内存等资源。还通过 adb 实现手机端与电脑的文件传输通过 adb 查看手机端 app 运行的日志，通过看日志来分析具体问题。

## 15.5 App 兼容性测试怎么做的？

像兼容性这块当时，我们主要用真机测试为主，公司当时使用提供大概 7、8 款机型吧，我记得像华为荣耀系列两款，例外小米机型有选择 2 款，还有就是像 vivo，oppo 当时都有测过，对了还有三星等这些系列机型上都有做过真机测试。

真机这块，像系统版本主要覆盖的系统其中 6.0\7.0\8.0 为主，5.0 以下公司当时都不要求测，

对于其他的机型覆盖不到位，我们都是通过云测进行覆盖的，云测这边，我们公司用的 testin 这个云测平台，公司有提供账号给我们只要登录上去，然后把 apk 上传上去，之后选择机型要测试的机型，当时我们在云测测试有差不多有 60 多款机型吧，主要是市面上流程的主流机型，每个系列都会选个几款，如果用真机测了的就不在选择了，然后做一些相关的配置，云测平台上主要帮我们做了智能遍历，安装，启动，运行，卸载，初始化，Monkey 测试相关的测试，不过 monkey 一般都是通过真机测的，云测平台没有测过，配置好了之后，提交测试就可以了，一般提交测试之后，需要几个小时就会出报告。然后分报告，主看遍历，安装，启动，运行，卸载，初始化相关哪些机型有出问题，对于出问题的机型，一般会先补测一下，如果还有问题，我们项目组一般会向公司申请真机再真机进行复测，如果真机复测有问题，就通过利用 adb logcat 查看错误日志，分析具体的问题所在。

其实我们做兼容性测试主要就是看软件在不同机型，不同系统版本下能不能正常安装，卸载是否能正常启动，运行，初始化，我们都把各个功能都进行运行一遍，主要就是跑下主流程，看有不有问题。例外，就是看软件在不同屏幕大小，不同的分辨率的手机下显示是否正常，有不有拉伸，显示不全，或者显不清晰的等问题。

当时我们兼容性就这么做测。

## 15.6 App 稳定怎么做的？Monkey 怎么用 (App 稳定测试)？

稳定性这块，我们当时用的是 SDK 自动的一个 Monkey 工具进行测试的，其实 Monkey 工具主要通过模拟用户发送伪随机时间去操作软件，通过执行 Monkey 命令，它会自动出报告，执行测试大概在 10 万次，每个动作的间隔时间 250ms，主要就是看软件长时间，随机乱操作的情况，是否会出现异常，闪退，崩溃等现象。

一般我都是在下班的时间晚上时间执行 Monkey 命令，并把生成的报告导出到电脑端，大概需要 6、7 小时，第二天早上看报告，分析报告，如果出现问题，一般利用上次执行的那个种子值，再进行执行命令进行复测一下，

像 monkey 命令：

```
adb shell monkey -p com.xy.android.junit -s 种子值 --throttle 250 --ignore -crashes --ignore -timeouts -monitor-native-crashes -v -v 100000 > E:\monkey_log\monkey_log.txt
```

这里主要关注几个点，1、指定种子值 2、忽略一些异常，保证能正常执行完成



3、设置间隔时间 4、配置一些时间比例 5、然后就是执行的次数。

对于报告怎么分析这块，主要看有木有 CRASH(崩溃)，ANR(超时无响应)，Exception(异常)等的情况像看有木有空指针异常(NullPointerException)啊，OOM 等现象啊等等，

找到 CRASH 崩溃 ANR 超时无响应 Exception 异常的位置，看出现错误的上一个动作是什么，什么做了什么动作导致错误出现。异常信息会详细的指出哪个 Activity 出现了问题，甚至于哪个函数出问题了，具体哪个位置。然后把报告中出现的日志信息截图发给开发，开发修复完成之后，我们会根据种子值在进行复测一下。

稳定性这块我们当时就是这么做的。

我在运行 monkey 发现过很多的问题我就简单的说几个问题，举个例子在查看 monkey 运行日志中

(1) `com.androidserver.am.NativeCrashListener.run(NativeCrashListener.java: 129)`

属于本地监听代码导致的崩溃，手机监听代码导致的崩溃，他是手机产生的原因，不是代码的原因不做处理。

(2) `//Short Msg:java.lang.IllegalArgumentException 传参异常：需要一个 stng 类型，给我一个 int 类型`

(3) `//Short Msg:java.lang.NullPointerException 空指针异常`

(4) `//Short Msg:Native crash 本地代码导致的奔溃`

(5) `(com.koudaizhekou.tbk/com.uzmap.pkg.EntranceActivity) 超时无响应`

等等等……

## 15.7 App 弱网测试怎么做的？

弱网测试这块我用的 fiddler 工具做的，通过 fiddler 实现延迟发送数据或接收的数据的时间来限制网络的下载速度和上传速度，从而达到模拟 2G\3G\4G 的移动网络的弱网场景，还有设置随机数来模拟网络不稳定的情况。

操作：首先保证手机与电脑在同一个网络，然后在手机上，设置代理服务器，指定服务器为装了 fiddler 的电脑的 ip 地址，端口为 8888 然后就是在 fiddler 上设置上行，下行速率，实现对发送，接受数据的进行网络延迟具体在 fiddler 的菜单上有一个 Rules -> CustomizeRules，打开 Fiddler 的 ScriptEditor 文件，在其中找到 `m_SimulateMode` 标志位，然后修改上行，下载的网络延迟时间即可，具体设置参数的值 SE 那边有给到一个参考文档

2G 上行 440，下行 400， 3G 上行 100，下行 100

模拟网络不稳定的情况

生活当中，网络并不是固定不变，一般处于不稳定的状态，我们这个时候编写随机数来模拟网络不稳定的情况

1, 在 Customize Rules 文件中, 写一个随机的数

```
static function randInt(min, max) {  
    return Math.round(Math.random()*(max-min)+min);  
}
```

2, 在 Customize Rules 文件中, 限制网络的参数修改成随机数

```
if (m_Simulate Modem) {  
    // Delay sends by 300ms per KB uploaded. 2kb/s  
    oSession["request-trickle-delay"]=""+randInt(1, 20000);  
    // Delay receives by 150ms per KB downloaded. 2.5kb/s  
    oSession["response-trickle-delay"]=""+randInt(1, 20000);  
}
```

3, 重新勾选

Rues-> Performance->勾选 Simulate Modem Speeds

弱网测试, 看我们软件在弱网场景下是否会有丢包的现象, 丢包率是否严重, 页面是否能正常展示, 是否有空白页, 数据是否有丢失, 页面加载速度是否会严重影响用户体验。

## 15.8 App 的性能测试

内容要点:

指标: cpu, 内存, 电量, 流量, FPS,

怎么测? cpu, 内存, 流量 android studio

cpu 不能超过 10-20% 普通业务要求在 10%左右, 核心的业务, 尤其是一些绘图的业务,

要求在 20%左右。指标: SE 给到, 竞品分析

内存主要看有内存泄露的情况, 怎么看?

流量: 一刷新使用软件, 流量会逐渐增加

具体操作:

1, 对于 cpu 内存, 流量这 3 个指标, 我们用的 android studio 来检测的, 结合 sdk 里面的一个插件 android Monitor, 它可以帮我们试试检测 cpu, 内存, 流量的曲线图

2, 这里我们需要开启手机端的开发者模式, 还有 usb 调试模式, 例外, 开发给我们提供的 apk 包, 必须需要开启 debug 模式, 开发开启之后打包给测试人员就可以了

3, 主要就是对我们需要测的功能进行操作, 然后实时查看图表, 看他是否有内存 oom 内存泄漏, cpu 是否使用过高, 是否有内存抖动, 造成的卡顿等现象, 是否有图片过大造成流量使用过大这些问题等, 是否有频繁使用流量, 没有使用缓存等问题

4, 像 fps 帧率是通过 adb 命令来测的, 还有电量我们当时用的是手机自带的一个第三方软件测的?

## 15.9 对于内存具体怎么测呢?

对于内存其实主要看有没有内存泄漏的问题

具体我们是这样做的:

1、首先我会频繁操作同一个业务,看他的内存和 cpu 是否逐步增长,最后稳定在一个固定大小的范围,如果在频繁操作同一个业务,内存一直在增长,可能存在有内存泄漏问题,尝试手动 GC(手动回收内存,因为内存泄漏,系统已经回收不了,所以尝试下手动回收内存),内存明显或者断崖式的下降,基本就可以判断有内存泄漏的现象

再通过 `dump java` 这个去分析,分析结果如果出现 `leaked`,就说明有了,里面可以找到是哪个对象,截图提 bug,

2、使用 app 过程中,内存一直在增长,那基本可以判断有内存泄漏的情况,还有看是否有内存抖动的现象:这里主要原因还是有因为大量小的对象频繁创建,频繁的回收内存,会导致 cpu 频繁使用,造成 cpu 使用过大,造成 app 卡顿,导致内存碎片,内存泄漏等问题

## 15.10 对于 CPU 具体怎么测呢?

cpu 主要就是看有没有过高,有没有超过我们的指标范围

具体是这样做的:

首先频繁使用某一个业务,cpu 是否逐步增长,最后稳定在一个固定大小的范围,对于一把基础业务,对 cpu 要求不高的业务,cpu 不能超过 10%,对于 cpu 要求比较高的,比如某个业务需要加载地图,大量的图片,视频等的业务,或者需要做大量的数据统计分析的业务,我们要求 cpu 不能超过 20%

## 15.11 对于流量具体怎么测?

a,首先看在没有操作功能业务的情况下,没操作流量不应该有,或者是流量使用不是很大,就几 KB 因为 app 肯能实时刷新消息,比如如果一个登陆,你就使用 1M 的流量,查询个图片使用 3-4M 的流量图片,这个肯定流量使用过大

b,频繁操作同一个业务,流量一直在刷,说明没有使用缓存

如何处理:图片过大处理方法:图片压缩传输,要么降低图片分辨率,

## 15.12 对于 FPS 具体怎么测?

对于 Fps 帧率的问题,我们当时用的 adb 命令来测的

知识点拓展: Android 设备的屏幕刷新率为 60 帧/s,要保持画面流畅不卡顿,要求每一帧的时间不超过  $1000/60=16.6\text{ms}$ ,这就是 16ms 的黄金准则,

a. 打开手机:开发者选项-> profile GPU rendering ->

in adb shell dumpsys gfxinfo(开启 GPU 渲染模式)

b. 操作要测试的 apk

C. cmd 窗口输入命令: adb shell dumpsys gfxinfo 包名

d. 得到一个矩阵数据, 计算矩阵中帧率大于 16 的点所占比例, 即为卡顿比

e. 通过 excel 进行表格处理可以直观的查看软件的流畅度

15.14 对于电量具体怎么测?

电量这一块, 我们当时用的手机自带的第三方软件测的

### 15.15 App 交互性怎么做的?

交互性这块, 主要从以下几个方面去考虑测试的是:

1. 看我们软件与其他应用软件的同时运行来回切换是否有问题

2. 看软件切换到后台一段时间, 再切换到前端, 或者前后台来回切换,

软件是否会有异常, 比如: 进程被杀死, 或者切换到前端页面出现问题, 或者页面数据丢失等等。

3. 看软件被在使用过程中被其他应用中中断, 或者其他意外情况中断, 比如: 来电, 来短信, 闹铃, 低电量测试等, 还要注意手机端硬件上, 如: 待机, 锁屏, 插拔数据线, 耳机等操作不会影响客户端。

### 15.16 App 的安装, 卸载, 更新测试具体从哪些方面考虑?

**安装测试:**

1. 正常安装测试, 检查是否安装成功安装完成后, 能否正常启动应用程序

2. 是否支持第三方安装, 比如豌豆荚及 91 助手等工具可以正常安装及卸载程序

3. 检测在各大手机市场上下载, 并直接安装, 看是否能正常安装, 安装完成之后, 能否能正常启动。

4. 检测 APP 版本覆盖测试(先安装一个低版本, 不卸载。然后再直接安装一个高版本, 看是否会覆盖低版本。(直接覆盖是否成功, 卸载之后, 再下载新版本, 看是否能安装成功)

5. 检测版本回退(先装高版本, 不卸载, 直接再重新安装一个低版本, 是否会覆盖高版本。

6. 检测在内存不足的情况下, 去安装软件, 系统应该会有提示

7. 在安装过程中, 所有的提示信息必须是英文或者中文, 提示信息中不能出现代码符号、乱码等。

8. 检测在未允许外来程序的安装的情况下, 安装软件是否会有提示信息

9. 如果软件需要用到网络, GPS 定位, 联系人等其他系统应用程序的时候, 安装 App 会有相应的提示。在不允许的情况, 软件是否能正常使用。

10. 安装完成后, 是否对其他应用程序造成影响。

11. 多进程进行安装, 是否安装成功。(同时安装多个软件, 是否能正常安装)

12. 在不同型号、系统的手机进行安装。(兼容性)

13. 安装过程中，取消安装，再次重新安装，是否能正常安装。
14. 安装完成后，检查手机桌面能否添加快捷方式。（是否有快捷图标生成。）

#### **卸载测试：**

1. 正常卸载，软件是否能正常被卸载，相应的桌面图标是否会删除
2. 卸载之后，对应的文件是否删除干净，进入安装位置，去看下是否有残留的文件
3. 程序正在运行的时候，卸载程序，是否能正常卸载
4. 卸载过程中，取消卸载，看是否正常退出卸载程序，检查软件是否还能继续正常使用
5. 在没用使用程序时，删除目录文件，看程序是否能运行。
6. 不同系统、硬件环境下进行卸载
7. 卸载成功后，是否对其他程序有影响
8. 卸载后再次安装，是否正常使用
9. 在卸载过程中，所有的提示信息必须是英文或者中文，  
提示信息中不能出现代码是否有相关的提示信息
10. 卸载过程中，出现意外(比如手机关机，没电，查看信息，接打电话)，程序是否还能运行
11. 在卸载过程中，突然重启设备，再次访问程序，是否还能运行。

#### **更新测试：**

1. 当客户端有新版本时，提示更新
2. 非强制更新，可以取消更新，旧版本正常使用，下次使用软件时，仍然会出现更新提示
3. 强制更新，强制更新而用户没有更新时，退出客户端，下次启动，依然提示更新
4. 不卸载，更新，检查是否可以更新
5. 不卸载更新，检查资源同名文件如图片等是否更新成最新版本。
6. 非 wifi 网络下，提示是否更新，取消就加入待下载，wifi 下自动更新。

### **15.17 H5 界面怎么测试**

基本功能测试：（浏览器、微信内置浏览器）

#### **登陆**

目前 H5 与 native 各个客户端都做了互通，所以大家在测试的时候要注意两点：

- A、若客户端已登录，那么进入 H5 后仍然是登录状态
- B、若客户端未登录，进入 H5，点击对应按钮 OR 链接，如果需要登录，须拉起 native 登录。若取消登录，是否可再次拉起登录，或者停留在的页面是否有对应的登录提示。。

ps：本次测试过程中就发现，第一次点击链接，可以拉起登录，第二次却不能

#### **翻页**

遇到翻页加载的页面，需要注意内容为 1 页或者多页的情况。

A、数据分页加载时，注意后续页面请求数据的正确。

ps：这个需要注意在快速操作场景中，请求页数是不是依次递增，快速操作(如第 1 页尚未 loading 出来的时候仍然继续上拉操作)时是否发出去对应的请求了。

### 刷新与返回

A、下拉刷新是否仍然处于当前页面。

B、用户主动点击刷新按钮是否仍然处于当前页面

C、点击返回与 back 键，回退页面是否是期望页面

ps：本次测试过程中就发现，mtop 接口请求成功，但是 data 内无数据时，返回到的就是个空白页面，无法正常发送请求。

### H5 适配相关

H5 的适配其实比客户端的相对来说，要少一些，手机品牌之间的差异不大，所以不用太多关注，最容易出现的是 android2.3 系统，这个要特别关注下：

A、大屏(如 720\*1280，重点关注页面背景是否完全撑开页面，刷新是否有抖动)

小屏手机(如 320\*480，重点关注下弹框样式和文案折行)

B、android2.3、android4.X 随机找一个即可

C、ios5、ios6、ios7

### 体验相关

### 资源相关

A、页面中有图片的话，淘宝那边建议图片一般不大于 50kb，本着一个原则，尽量缩小图片

B、资源是否压缩、是否通过 CDN 加载

C、如何保证二次发布后有效更新。

### 流量

A、对于一些不会变化的图片，如游戏动画效果相关图片，不需要每次都请求的东西，做本地缓存

B、数据较多时是否做了分页加载。

### 页面展现时间

A、关注页面首屏加载时间

### 页面提示

A、弱网络下，数据加载较慢，是否有对应的 loading 提示

B、接口获取异常时，提示是否友好

C、刷新页面或者加载新内容时页面是否有抖动

## 手机操作相关

A、锁屏之后展示页面

B. 回退到后台之后，重新呼出在前端展示

### 15.18 你们之前是用什么手机什么版本做兼容性测试的？

有用到三星 note5 Android 6.0.1      三星 s6 Android 6.0.1

红米 1s Android 5.1      小米 5 Android 7.0

华为 mate9 Android 6.0      乐视 2 Android 6.0

华为 mate20 Android 9.0      三星 s8 Android 8.0

iphoneos ios 10.3.2      iphone ios 10.0.2

iphone ios 8.4.1      iPhone X ios 11.0

### 15.19 Android 跟 ios 测试有什么区别？

Android 和 ios 测试的共同点都需要进行界面测试、功能测试、兼容性测试、网络测试、交互性测试、易用性专项测试、异常测试、安全专项测试以及权限测试。不同的是 Android 测试除了以上的测试之外还要用 monkey 进行稳定性专项测试以及用 emmagee 或者 gt 进行性能专项测试。ios 是用 itools 工具对功能进行测试：安装、传输文件以及查看日志。

从操作系统，安装卸载，按键操作，开发语言这几个方面去区分操作系统： android 操作系统较多，iOS 较少只能升级不能降级，并且新的版本的资源库不能完全兼容旧版中系统中的应用，如果低版本应用调用了高版本的资源库，可能会导致系统崩溃。

安装卸载测试，应用发布后：下载安卓包的平台和渠道很多：豌豆荚、应用宝、360 手机助手等；

iOS 主要有 App store、iTunes，安全性会更高点

本地测试：安卓手机可以通过扫码或者直接安卓 APK 包安装测试包；iOS 要安装测试包必须绑定手机的 id(证书)才可以安装 ipa 测试包

按键操作测试：安卓手机针对每一款手机有不一样的操作；苹果手机操作习惯单一

开发语言：虽然同样的业务安卓和 iOS 的展示形式和业务一致，但是底层全完不一样，安卓的应用是有 java 语言实现的，iOS 用 OC 实现。

### 15.20 小程序怎么测试

1，小程序测试(多用第一人称，口语化表达，多讲一些，讲细一些，先宏观，在微观)参考面试问题 STAR 法则：

我们原来主要测试，几个方面，界面测试，功能测试，交互性测试，兼容性测试，安全测试，易用性测试，异常测试，权限测试

界面测试：主要是测试跟界面的原型图是否一致，同时我也要考虑不同屏幕大小跟分辨率

功能测试：跟所有的功能测试都是一样的，还有小程序有位置功能，检查下，微信小程序,附近中是否能找到对应小程序，使用小程序是否记录，

交互性测试：要考虑跟微信的功能交互使用，比如说一些，卡包，支付等功，考虑跟手机固有功能交互，比如说来电，短信等

兼容性测试：考虑跟微信不同版本的兼容，还有同时还要考虑不同手机厂商跟手机型号兼容，还要考虑当微信清除缓存后，小程序还能否继续使用

安全测试：测试数据加密，包括 sql 与 xss 脚本攻击这块

易用性测试：考虑功能是否方便还用

异常测试：考虑断网，手机重启，关机的情况

权限测试：小程序继承微信权限，测试手机对微信权限，还要考虑微信对小程序授权，是否允许操作原来我们测试阶段，上传小程序到微信小程序平台，上传到开发版本里面，通过扫描二维码去下载小程序进行测试，

上线后，我们也要测试下，微信搜索小程序中能否搜索的到

## 15.21 公众号，小程序比 app 更火，你怎么看

小程序有安装包，一般控制在 1M 以内

- 1，基于微信大的平台，有流量的入口
- 2，不需要安装，操作更加方便

## 15.22 微信开发者工具如何使用

原来我们的下载一个微信开发者工具，导入开发给的小程序代码包，在输入开发给予的 appid，调试与测试小程序代码包，

如果真机测试，也可以扫描开发在微信开发者工具生成的二维码进行测试

# 十六、安全测试

## 16.1 安全测试怎么测试?

安全测试我们常用的方法是：1，sql 注入 2，xss 脚本攻击 3，数据加密 4，权限控制

运用扫描工具 appscan 扫描 web 系统。扫描 app 的一般用腾讯 wettest 等平台测试测试，

前端(web)时我们首先检查一下用户的敏感信息有没有进行加密显示,通过 Fiddle 抓包工具检查一下用户的敏感信息有没有进行加密后传输如：用户密码，相应的银行卡，个人信息等，再到日志中搜索关键信息，搜索到，就泄密，存在安全漏洞，数据库中是否做了加密处理。还有就是把发送请求的数据篡改例如：打开 fiddle 工具，设置过滤器，设置断点把商城里的支付订单的 1000 完金额修改



成 1 块钱，再放行发送数据，看是否，支付 1 块钱，订单支付成功，如果成功则是 bug，预期结果是支付金额不对才行，用 sql 语句注入和 xss 脚本攻击，把 sql 语句或 xss 脚本编辑成 csv 文档通过 jmeter 工具添加“CSV 数据文件设置”来调用里面的语句来在所有的查询位置、所有的接口请求、url 请求链接参数中，进行 sql 语句注入或 xss 脚本攻击：模拟 sql 语句 xss 脚本攻击数据库，看它是否把 sql 注入代码或 xss 脚本当成字符处理，如果不是就是 bug，再看功能不能正常使用、系统是否崩溃，错误弹出框，界面是否变形。还有在所有可以保存的地方，都添加 xss 脚本攻击代码，看看是否出现异常。权限控制(用户的校验)校验用户安全认证信息是否正确，如没有权限的界面不能进入，不能提供操作入口或不能通过 url 直接去访问：例如;把所有需要权限的界面，全部复制一份出来然后退出用户再通过 url 链接去访问功能如果可以访问，就是 bug，如有提示登录，属于正常。还有登陆用户再退出用户后，点击浏览器的后退不能进入原来的界面。

通过 appscan 扫描工具去扫描，我们般是这样操作的；

- 1、启动软件进入主界面->选择创建新的扫
- 2、在弹出的新建扫描对话框中选择常规扫描；
- 3、在弹出的扫描配置向导对话框中选择 AppScan(自动或手动)，点击下一步；
- 4、在此页面中填写需要扫描系统的同址，点击下一步；
- 5、选择登陆方式为记录，点击下一步；
- 6、完成系统登录操作后，关闭浏览器选择登录管理中的“详细信息”；
- 7、在“详细信息”中，取消“激活会话中的检测”，同时在“登录会话标识中”，对变量参数值进行跟踪以确保参数的正确处理而获得系统访问权限；
- 8、把密码修改页面 URL、用户权限删除 URL 等页面添加至排除测试范围，防止 Appscan 的请求产生相应影响(防止 Appscan 修改密码或删除用户帐号)；
- 9、在“自动表单填充中，填入用户名和密码参数，并填写参数值，其余不做修；
- 10、如果测试过程中，应用服务出现性能响应过慢的问题，适当调整测试线程数为 2~5；
- 11、测试策略选择为严重性”进行过滤只勾选关注的严重性级别，把“高”、“中”和低”选上，
- 13、点击确定按钮，结束扫描配置工作；
- 14、点击扫描按钮，选择“仅探索”， Appscan 开始执行探索工作；
- 15、探索自动结束后，点击“手动探索”按钮，此时，在弹出的 E 浏览器窗口中登录系统，手工的任意访问系统页面(1-10 个)，关闭浏览器；
- 16、在弹出的参数添加窗口中，点击确定按钮”
- 17、点击扫描按钮，选择“完全扫描”， Appscan 开始执行测试工作；
- 18、测试执行结束后，点击报告按钮，选择关注的测试结果信息，勾选“报告内容”里的所有勾选；

19、保存报告，保存类型建议选择为html;然后分析报告。(只需了解面试尽量不要提及)

## 16.2 appscan 的原理是什么

对于一个综合性的大型网站来说，可能存在成千上万的页面。以登录界面为例，至少要输入用户名和密码，即该页面存在两个字段，当提交了用户名和密码等登录信息，网站需要检查是否正确，这就可能存在一个新的检查页面这里的每个页面的每个参数都可能存在安全漏洞，可能成为被攻击对象。AppScan 正是通过按照设定策略和规则，对 Web 应用进行安全攻击，以此来检查网站是否存在安全漏洞。

在使用 AppScan 的时候，通过配置网站的 URL 网址， Appscan 会利用“探索”技术去发现这个网站存在多少个目录，多少个页面，页面中有哪些参数等，即探索出网站的整体结构通过“探索”可确定测试的目标和范围，然后利用 AppScan 的扫描规则库，针对发现的每个页面的每个参数，进行安全检查。

简言之， APPScan 的工作原理如下：

- 1) 通过“探索”功能，利用 Http Request 和 Response 的内容，爬行出指定网站的整个 Web 应用结构。
- 2) AppScan 本身有一个内置的漏洞扫描的规则库，可随版本进行更新，从探索出的 url 中，修改参数 or 目录名等方式，构造不同的 url 对照组向服务器发送请求 or 攻击。
- 3) 根据 Http Response 返回的内容，和正常请求所返回的响应作对比，是否产生差异性，而这种差异性又是否符合扫描规则库的设定规则，以此来判断是否存在不同类型的安全洞。
- 4) 若 APPScan 可判断存在安全漏洞，则对这些漏洞的威胁风险给出说明，进行严重程度提示，并给出修复的建议和方法，以及漏洞发现的位置提示。

## 十七、 python 编程

### 17.1 一个变量没有定义报什么错误

一般在 python 中会有红色的波浪线标出来要是运行后，就直接会报变量没定义的错

NameError: name 'i' is not defined

### 17.2 列表与元组的区别

他们最大的区别

- 1) 列表可以修改，元组不可以修改，列表就相当于一个动态数组，而元组是一个静态数组。
- 2) 从设计上说，列表是用来保存多个相互独立对象的数据集合，而元组设计的初衷就是为了描述一个不会改变的事物的多个属性。
- 3) 然后列表的声明用中括号元组的声明用小括号，并且元组只有一个元素的时候需要在后面加逗号。

### 17.3 python 序列有哪些?

有序序列: 字符串, 元组, 列表

无序序列: 字典, 集合

序列特点:

- 都可以通过索引得到每一个元素
- 默认索引值总是从零开始
- 可以通过切片的方法得到一个范围内的元素的集合
- 有很多共同的操作符(重复操作符、拼接操作符、成员关系操作符)

### 17.4 Python 区分大小写吗?

python 中是区分大小写的

### 17.5 如何获取字典中的所有键?

我们可以直接遍历这个字典, 用 `keys()` 函数取出所有的键

```
for key in dict.keys()
```

如果说要取出字典中所有的键和对应的值, 我们用 `items()` 函数

```
for key,value in dict.items()
```

### 17.6 列表和字典有什么区别?

- 1) `dict` 生成慢, 查找快, `list` 生成快, 查找慢,
- 2) 另外 `list` 是有序的, `dict` 是无序的,
- 3) `list` 通过索引访问, `dict` 使用 `key` 访问,

### 17.7 截取字符串里面部分字符怎么做?

- 1) 一般就是在定义的这个字符串变量后面用中括号 `[]`, 然后里面填写需要截取的索引数

```
print(str[2:1])
```

```
print(str[:2])    # 从开始到 2)
```

```
print(str[2:5])
```

```
print(str[:])
```

- 2) 利用字符串的中的内置函数 `split()`

```
str1 = '2018-11-20 14:40:00'
```

```
print("年份: " + str1.split(' ')[0].split('-')[0])
```

```
print("月份: " + str1.split(' ')[0].split('-')[1])
```

```
print("日期: "str1.split( ' ' )[0].split( '-' )[2])
print("时: "str1.split( ' ' )[1].split( ':' )[0])
print("分: "str1.split( ' ' )[1].split( ':' )[1])
print("秒: "str1.split( ' ' )[1].split( ':' )[2])
```

## 17.8 如何遍历列表，字典？

### 1) 列表遍历

```
list=[3,4,6,89]
for i in range(0,len(list)):
    print( '%d' %list[i])
for i in list
    print(i)
```

### 2) 字典遍历

```
dict ={ 'name' : ' jason' , ' pwd' : ' 123456' }
for key in dict.keys():
    print(dict[key])
for value in dict.values():
    print(value)
for key,value in dict.items():
    print(key,value)
```

## 17.9 列出 5 个 python 标准库？

os: 提供了不少与操作系统相关联的函数

sys: 通常用于命令行参数

re: 正则匹配

math: 数学运算

datetime: 处理日期时间

## 17.10 字典如何删除键和合并两个字典？

字典 dic1={ 'name' : ' laowang' , ' age' : ' 18' },

dic2=( 'name' : ' Xiaohua' }, 删除 dic1 中键名为 ' name' 的键，并合并 dc1 和 dic2

### 1) 删除键名为 name 的键

```
dic1={ 'name' : ' laowang' , ' age' : ' 18' }
```

```
del dic1[ 'name' ]
```

2) 合并两个字典，使用字典的成员方法 update()

```
dic2.update(dic1)
```

```
print(dic2)
```

### 17.11 python 实现列表去重的方法?

第一种方式:

```
#encoding = utf-8
```

```
import time
```

```
time_start=time.time()
```

```
print u"列表去重的七种方法"
```

```
print u"第一种测试方法"
```

```
repeat_list=[1,2,4,1,5,1,2,5]
```

```
result=[]
```

```
for i in repeat_list:
```

```
    if i not in result;
```

```
        result.append(i)
```

```
print u"第一种去重结果: ", result
```

第二种方法:

```
repeat_list=[1,2,4,1,5,1,2,5]
```

```
print u"第三种去重结果: ",list(set( repeat_list))
```

### 17.12 python 内建数据类型有哪些?

整型 -- int

布尔型 -- bool

字符串 -- str

列表 -- list

元组 -- tuple

字典 -- dict

### 17.13 列出 python 中可变数据类型和不可变数据类型，并简述原理?

1) 可变数据类型: 列表 list 和字典 dict;

2) 不可变数据类型: 整型 int、浮点型 float、字符串型 string 和元组 tuple

“python 中的不可变数据类型，不允许变量的值发生变化，如果改变了变量的值，相当于是新建了一个对象，而对于相同的值的对象，在内存中则只有一个对象，内部会有一个引用计数来记录有多少个变量引用这个对象；可变数据类型，允许变量的值发生变化，即如果对变量进行 append、+=等这种操作后，只是改变了变量的值，而不会新建一个对象，变量引用的对象的地址也不会变化，不过对于相同的值的不同对象，在内存中则会存在不同的对象，即每个对象都有自己的地址，相当于内存中对于同值的对象保存了多份，这里不存在引用计数，是实实在在的对象。

#### 17.14 如何将字符串转换为小写？

```
S = "What is Your Name?"
print(s.lower())    #返回小写字符串
print(s.upper())    #返回大写字符串
print(s.capitalize()) #字符串首字符大写
print(s.title())    #每个单词的首字母大写
print(s.swapcase()) #大小写互换
```

#### 17.15 回文数

```
num = input( 'Num:' )
    if num == num[::-1];
        print("回文数")
    else:
        print(不是)
```

## 二十、场景问题

### 20.1 有没有抓过包，为什么要抓包，具体怎么做的？

抓包这块，其实我们在工作用得是比较多的，一般我们都在测试过程碰到问题，我们有的候通过抓包去看一下前端与后台的数据交互，通过分析这个数据，来定位到底是前端问题，还是后台问题。主要用的就是 fiddler 工具，如果只是抓取 Web 端数据的话，只要在浏览器上设置代理服务器，指定代理服务器为本地的 fiddler 这个工具，设置好端口：8888 就可以了，如果要抓取手机端的包，首先需要保证手机与电脑(安装了 fiddler 的这个电脑)在同一个网络，在手机上要设置 http 代理服务器为手动，然后服务器指定为电脑的 ip 地址，端口设置为 8888

例外，如果要抓取 https 包，可能还需要在电脑上安装证书，手机端这边也要安装证书，差不多这样就可以。

## 20.2 为什么要进行抓包?具体怎么分析的?

主要是我们在测试过程中,出现 Bug 的时候,为了定位问题,分析问题,需要抓取前端后台的数据交互过程,通过看数据,从而来分析到底是前端问题还是后台问题

分析这块的话:

- 1、首先看发的请求是否有问题,请求的接口 url 是否有错误,参数是否有错误,如果 url 或传参有问题那就是前端 Bug
- 2、如果请求没有问题,看下后台返回的数据是否有问题,状态码 5 开头的基本都是后台问题,状态码为 200,响应数据与预期不一致,那也是后台 Bug,不过这里有可能是数据库问题,也有可能是后台代码问题,这里首先我一般会查看下数据库的数据,如果数据库没有问题,那就需要查看下后台的日志文件,具体分析下看是后台代码哪里出问题了,然后把日志信息截图给开发,
- 3、返回的数据没有问题,请求参数, u 也没有问题,那可能是前端代码是否转换有问题,那就是前端 Bug

## 20.3 大概给我讲下抓包的原理?

其实就是把 fiddler 设置为代理服务器,那么浏览器上发出所有请求都由 fiddler 这个工具来代理转发,其实就是说:通过 fiddler 来拦截客户端与服务端之间的数据交互,这样我们可以拦截到客户端给服务器发了什么数据,服务器给客户端响应了什么数据,然后对数据进行分析。

## 20.4 给你一个需求,你具体讲下,你怎么测试?

- 第一点:首先我要熟悉这个需求的背景,这个需要功能,我至少要知道这个需求做什么吧,做的是一个什么样的解决方法这个要搞清楚。
- 第二点:这些搞清楚,看这个需要和其他模块有哪些交互,需不需要关联起来分析,有没有影响到其他模块,或者数据有没有流向其他模块。
- 第三点:这些都清楚了,那么我会根据熟悉的情况看下工作量,哪个时间点交付,在这个时间点我需要做一个计划给老大说下,如果时间充分,一个人能搞定,那么就一个人搞定,搞不定,我会向老大申请人力。其实接下来也差不多了,分析需求,尽量要评审,在前期就保证好质量,如果没有问题。其他也就是用例实例化,执行啊,bug 跟踪啊等,最后出个报告就好了。
- 基本我做的活,会这么做主要看你在最短时间把需求吃透,做好计划,展开工作。

## 20.5 给你一个项目,你怎么展开工作,请详细说一下,你具体怎么做的

面试官考的点:进我公司,给你一个项目,你怎么做,看你有没有独立做项目能力

- 第一点:首先我要熟悉这个项目的背景,我至少要知道这个项目做什么吧,做的是一个什么样的解决方法这个要搞清楚

第二点：这些搞清楚，那么必须要清楚这个项目的核心模块和业务，我们给这个模块做一个优先级，先测试哪些核心模块，再做哪些模块

第三点：根据刚刚信息，我会熟悉这个业务，这个业务相关模块，以及写数据前后台交互等

第四点：这些都熟悉了，那么我会根据熟悉的情况看下工作量，哪个时间点交付，在这个时间点我需要做一个测试计划一般需求分析的话，根据我的经验基本 2 天差不多了，编写用例时间，执行时间，bug 修复时间，我会做一个工作量评估，如果还有其他测试，我会把这个计划发给大家，让大家评审这个计划和安排，看安排是否合理，大家能不能在这个计划时间段完成，基本需求分析等多给个 2 天左右，那么大家要再这个时间段完成，完成后我会安排大家讲解需求，做需求评审和需要讲解，前期时间多投入点，后面风险小一点，这个我主要跟进下进度就行。

第五点：那么接下来，大家按照计划去编写用例，用例在时间点完成后，我会组织大家评审，这个两个我必须把控，减少后面测试风险，前面分析和用例把握到位，后续执行和 bug 这块不会有太多问题后续的话，那么大家执行就行，我主要跟进这个进度，每天大家 bug 解决情况，以及测试进度，做成日报，像些问题不好解决的，有问题的，我去协助那么测试完成后，我会收集大家测试数据，做一份测试报告，这个我会评估这个测试有没有风险，没有的话，发送给开发老大，开发所有人，测试所有人。这块评审下，后续组织测试人员安排上线，我基本是这么做的

## 20.6 项目快上线因为开发延迟，测试时间被压缩怎么做？

[开发延期完成，产品还需如期上线，测试需怎么办

首先看下自己还剩下多少时间，然后要评估工作量，评估下风险，比如说，可能我根本做不完，风险太大，那么这个我首先要把风险提出来，给对应的领导，那么我会要求有人协助我完成，需要多少人力来完成，

如果只是说，正常时间完不成，那么我可以提下加班，看加班这个时间段能否完成，如果可以，我会和老大商量，提出加班计划。

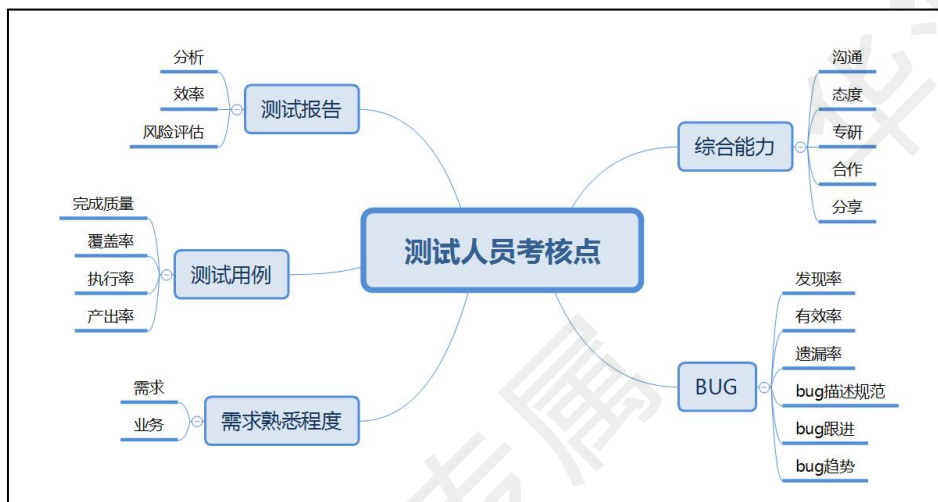
如果是其他的，先跟领导确认，比如说我只需要把优先级高的工作完成其他工作可以安排到下一个版本去做，那么我就做个计划，把计划个老大说下。

## 二十一、测试经理管理

### 21.1 员工的绩效考核，量化标准

- 1、考勤，任务完成率，失效，完成，满意度，用例编写量，提交 bug 量，线上问题
- 2、新的任务完成，新技术研究，对团队的贡献





<https://blog.csdn.net/maniacfish/article/details/6221148>

## 21.2 你组内有一个技术比你牛逼年轻的组员不服你，你会怎么处理

不要开除他，首先要认可他的能力，跟员工沟通，自身要有宽容之心，看下自身的不足以及需要提升的方面，技术方向，管理的能力等等，看下他为什么不服的，尽量做到让他服你

## 21.3 你的组员提交的 bug 开发不认同而吵起来了，你怎么处理

如果是我的组员跟开发因为提交的 bug 争吵的时候，我首先会先稳定下他们的情绪，先跟测试沟通为什么吵起来，沟通的时候要先扬后抑，不好的地方的话指出来。确定问题这边具体是什么原因，如果是测试这边的问题，就让测试员把自己的问题处理掉，当然，开发这边也是需要沟通的，说明争吵是不对的，主要是将问题说明清楚，如果是测试问题可以说明测试这边的问题，如果是开发这边的问题，我会指出开发错误的地方。