

软件测试工程师面试题系列篇 | 目录

1. 测试常见问题与流程篇
 2. 测试工具篇
 3. 计算机网络知识篇
 4. 数据库篇
 5. Linux 篇
 6. Python 编程篇
 7. 自动化测试篇：包含 Selenium、Appium 和接口测试
 8. 性能测试篇
 9. 软素质篇：10 大灵魂拷问
 10. 反问面试官篇
-

一、测试常见问题和流程篇

1. 介绍一下测试流程（重点，常见！）

- 需求评审、测试计划、测试用例、用例评审、冒烟测试、测试执行、验收测试、风险评估、上线观察、问题跟进、测试报告、复盘会议；
- 根据自己的日常经验来回答，每个点的工作内容都需要清晰掌握，有可能就某个点如何工作进行提问。

2. 介绍一下测试方法

- 按阶段：单元测试、集成测试、系统测试、验收测试
- 按手段：黑盒测试、白盒测试、灰盒测试
- 其他：冒烟测试、回归测试

3. 介绍一下测试用例设计方法（用例设计方法&测试方法需分清楚）

- 黑盒测试用例设计：等价类划分法、边界值分析法、错误推测法、因果图法、正交试验分析法、流程分析法
- 白盒测试：语句覆盖、判定覆盖、条件覆盖、条件组合覆盖、判定/条件覆盖、路径覆盖

4. 设计一个登录页面的用例（提供某个场景设计用例，重点！）

- 功能测试：正确输入、为空输入、字符类型校验、长度校验、密码是否加密显示、大写提示、跳转页面是否成功、登出后用另一个账号登录
- UI：界面布局合理、风格统一、界面文字简洁好理解、没有错别字
- 性能测试：打开登录页面需要几秒、点击登录跳转首页需要几秒、多次点击、多人点击
- 安全性：用户名和密码是否加密发送给服务器、错误登录的次数限制（防止暴力破解）、一台机器登录多个用户、一个用户多方登录、检查元素能否看到密码
- 兼容性测试：不同浏览器、不同的平台（Windows、Mac）、移动设备能否工作
- 易用性：输入框可否tab键切换、回车能否登录等

5. 举例说明项目推进的能力（针对个人评价的举例说明）

- （例）推动开发解决菜单权限需退出登录才可应用的问题。
- 描述：XX项目上线后，由于权限更新导致用户无法使用旧界面+用户不懂得自己退出登录以应用新菜单权限，线上多个用户反馈平台相关功能无法使用。处理：远程操作+线上指导出现问题的用户退出登录，凌晨脚本批量强制用户退出登录。推进：在下一次版本中，推动开发处理菜单权限更新问题，以防每次更新菜单都出现用户使用不了功能的问题。结果：处理为用户登录期间仍可使用旧界面，直到用户退出重新登录后，才应用新菜单权限；
- 推动冒烟测试：冒烟不通过，测试召开会议罗列项目不通的模块、存在的问题，——对应到每个人去跟进，得到解决的时间，后续项目群说明并艾特每个人跟进。测试准时验收。
- 推动文档质量：在日常工作中遇到需求文档、设计文档、接口文档不规范或不详细的在绝大多数，这个时候就要通过沟通或以bug的形式，促使各个岗位将各自的文档完善。
- 结合自己的个人经验，从问题描述+处理过程+推进表现+结果，——说明。

6. 测试中遇到的比较难的一个项目是？（掌握自己简历上的项目）

- XX项目：这个项目是一个新项目、需要对接第三方、从协助测试产品模块到转手负责整个项目加强了一定的需求理解难度、订单逻辑不熟悉、其中需求的问题推进比较困难。
- 针对这些困难，测试方面：先快速了解需求，并与原先的测试负责人了解具体的进度和需要注意的测试点。开发方面：迅速与开发人员了解订单逻辑，以及与第三方对接的过程。产品方面：遇到某个需求问题产品不接受、最后汇报领导+开会讨论了较好的处理方案。虽然过程问题比较多，但是通过项目组的集体努力，最后项目得以按时上线
- 说明项目（最好是简历上的项目方便面试官进一步提问）。描述有难度的地方+如何处理+结果

7. 印象深刻的bug？

- 隐藏得比较深的bug、影响比较大的bug、处理过程比较曲折的bug。根据自己的经验描述：如何发现的、如何处理、影响、结果、反思。
- 举例说明：如升级版本兼容性问题、接口安全性问题、数据库安全性问题、服务器资源占用溢出问题、代码逻辑问题等

8. 你们公司是不是敏捷开发？介绍一下敏捷开发？

- 是，敏捷快速迭代、多版本同时迭代
- 敏捷开发属于增量式开发，对于需求范围不明确、需求变更较多的项目而言可以很大程度上响应和拥抱变化、主张简单、拥抱变化、可持续性、递增的变化、高质量的工作、快速反馈、软件是你的主要目标

9. 复盘会议的主要内容有哪些？

- 这点需要结合自己平时参与的项目会议举例说明。如线上bug分析、优化改进策略、bug优先级等等

10. App 的兼容性怎么测，App 的接口测试怎么测？

- 系统兼容(ios、安卓)、机型兼容（iPhone、华为、小米、三星、vivo、OPPO）、分辨率兼容、软件本身向前向后兼容
- 接口测试：获取接口文档，使用fiddler抓包工具获取接口的请求方式、url、请求参数、返回参数，然后使用postman、jmeter进行测试

11. Web 端测试和 App 端测试有何不同（常见）

- 系统结构方面
 - Web 项目，b/s架构，基于浏览器的；Web 测试只要更新了服务器端，客户端就会同步会更新；
 - App 项目，c/s结构的，必须要有客户端；App 修改了服务端，则客户端用户所有核心版本都需要进行回归测试一遍；
- 兼容方面
 - Web项目：a. 浏览器（火狐、谷歌、IE等）b. 操作系统（Windows7、Windows10、Linux等）
 - App项目：a. 设备系统: iOS（ipad、iphone）、Android（三星、华为、联想等）、Windows（Win7、Win8）、OSX（Mac）b. 手机设备可根据手机型号、分辨率不同
- 性能方面
 - web项目 需监测 响应时间、CPU、Memory
 - app项目 除了监测 响应时间、CPU、Memory外，还需监测流量、电量等
- 相对于 Web 项目，APP有专项测试
 - 干扰测试：中断，来电，短信，关机，重启等
 - 弱网络测试（模拟2g、3g、4g，wifi网络状态以及丢包情况）；网络切换测试（网络断开后重连、3g切换到4g/wifi等）
 - 安装、更新、卸载
 - 安装：需考虑安装时的中断、弱网、安装后删除安装文件等情况
 - 卸载：需考虑 卸载后是否删除 App 相关的文件
 - 更新：分强制更新、非强制更新、增量包更新、断点续传、弱网状态下更新
- 界面操作：关于手机端测试，需注意手势，横竖屏切换，多点触控，前后台切换
- 安全测试：安装包是否可反编译代码、安装包是否签名、权限设置，例如访问通讯录等
- 边界测试：可用存储空间少、没有SD卡/双SD卡、飞行模式、系统时间有误、第三方依赖（QQ、微信登录）等
- 权限测试：设置某个 Ap
- p 是否可以获取该权限，例如是否可访问通讯录、相册、照相机等

测试面试之测试工具考点

1. 介绍一下测试中常用的工具（必备基础，必须掌握！）

- 需求问题跟进、测试计划、风险评估登记、测试报告、复盘会议：wiki
- 测试用例：Xmind 编写，Testlink 管理
- 测试执行：ELK、Xshell等
- Bug 管理：Jira、bugfree、禅道等
- 接口测试相关：Charles、Fiddler、Postman、JMeter等
- 自动化相关：Selenium、Appium、pytest、Locust、JMeter等

2. 用什么工具对用例进行管理？

- Testlink 管理用例的一般步骤：新建计划、新建版本、上传xml文件、添加测试用例到测试计划中、分配测试用例给开发、查看用例执行报告。
- Xmind：（略...）
- Excel：（略...）

3. 怎么使用 ELK 定位日志？

- 查看产品推送是否成功。产品从 A 平台推送到 B 平台，根据 A 平台的链接 id，搜索对应的日志。搜索不到，则为 A 平台推送失败。搜索到了，查看推送的状态，进一步判断问题所在。
- 使用官方文档，可以进一步了解 ELK 日志查看。

4. Xshell如何登录，如何切换目录？

- 使用 SSH 密钥登录：生成密钥公钥和私钥-上传公钥到服务器-配置 Xshell 使用密钥认证方式登录到服务器；
- 参考文章：<https://www.cnblogs.com/Black-rainbow/articles/9418713.html>
- 使用账号密码登录：配置中输入被连接服务器的账号、密码、IP 及端口连接；

5. 埋点测试怎么测试，使用什么工具，数据要不要入库？

- 使用 Charles、Fiddler 抓包，查看对应的来源记录、事件等必要参数是否正确，查看数据库记录是否正确

6. 介绍 Fiddler 和 Postman 的区别

- Fiddler 主要是抓包，Postman 主要进行接口请求；

7. 怎么使用 Postman 进行多个接口请求？

- 将多个接口请求归纳到一个集合里，在集合的右上角点击展开箭头，点击 Run；

8. 日常工作中 JMeter 是怎么用的？

- 接口测试：通过对指定接口进行请求访问，验证数据出入的准确性与安全性；
- 性能测试：编写对应的测试集，通过脚本控制线程数，实现逐步加压等；
- 结合自己项目经验，没有经验千万不能盲目举例乱说，避免坑自己；

9. 例举熟悉的自动化测试工具，并说明其实现原理

- 调用 Android adb 完成基本的系统操作
- 向 Android 上部署 BootStrap.jar
- BootStrap.jar Forward Android 的 4723 端口到 PC 机器上
- PC 上监听端口接收请求，使用 webdriver 协议
- 分析命令并通过 forward 端口发给 BootStrap.jar
- BootStrap.jar 接收请求并把命令发给 UIAutoMator
- UIAutoMator 执行命令
- 运行用 Python 写好的 Selenium 脚本，它会像 Web Service 中发送一个 HTTP 请求；
- 浏览器驱动中的 Web Service 会根据这个请求生成对应的 JS 脚本，因为不同的浏览器，相同的操作生成的 JS 脚本会有所不同，因此不同的浏览器要有不同的驱动；
- JS 脚本驱动浏览器，产生各种操作，并返回给 Web Service；

- Web Service 将结果通过 HTTP 响应的形式返回给客户端；

计算机网络篇（基础知识）

1. 擅长哪些开发语言？

- 学习过 Java , C 等
- 半精通 Python

2. 输入 URL 到网页显示出来的全过程

- a. 输入网址
- b. DNS解析
- c. 建立tcp连接
- d. 客户端发送HTTP请求
- e. 服务器处理请求
- f. 服务器响应请求
- g. 浏览器展示HTML
- h. 浏览器发送请求获取其他在HTML中的资源。

3. HTTP 和 HTTPS 的区别

- HTTPS 里面是要有证书的，HTTP 并没有证书。证书的作用是证明你是这个网站的拥有者。谁去证明？最顶级的 CA 去帮你证明，这些顶级的 CA 都是浏览器、操作系统本身就自动帮你集成，而且自动添加到设置信任里面去；
- HTTPS 要兼顾安全+性能的方面，由于对称式加密虽然速度很快，但是安全性特别的低，因为双方要规定对称式加密的密钥，别人都无法知道，但你怎么能确保别人不知道你的密钥呢，因此需要有非对称式加密去保证安全，但非对称式加密速度又很慢，如果客户端和服务端都用非对称式加密，网络得卡死了。所以当双方建立好了非对称加密后，再约定一个随机数，等大家都非对称解密了之后呢，就拿到只有对方知道的唯一随机数（密钥），就可以用密钥来进行对称式加密和解密了；

4. HTTP 的报文结构

- HTTP请求报文：一个HTTP请求报文由请求行、请求头部、空行和请求数据4个部分组成
- HTTP响应报文：HTTP响应也由三个部分组成，分别是：状态行、消息报头、响应正文

5. HTTP 常见的响应状态码

- 200 请求已成功，请求所希望的响应头或数据体将随此响应返回。
- 201 请求已经被实现，而且有一个新的资源已经依据请求的需要而建立，且其 URI 已经随 Location 头信息返回
- 202 服务器已接受请求，但尚未处理
- 301（永久移动）请求的网页已永久移动到新位置。服务器返回此响应（对 GET 或 HEAD 请求的响应）时，会自动将请求者转到新位置。
- 302（临时移动）服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进

行以后的请求。

- 303（查看其他位置）请求者应当对不同的位置使用单独的 GET 请求来检索响应时，服务器返回此代码。
- 304（未修改）自从上次请求后，请求的网页未修改过。服务器返回此响应时，不会返回网页内容。
- 305（使用代理）请求者只能使用代理访问请求的网页。如果服务器返回此响应，还表示请求者应使用代理。
- 307（临时重定向）服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。
- 401 当前请求需要用户验证。如果当前请求已经包含了 Authorization 证书，那么 401 响应代表着服务器验证已经拒绝了那些证书
- 403 服务器已经理解请求，但是拒绝执行它。与 401 响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交
- 404 请求失败，请求所希望得到的资源未被在服务器上发现
- 500 服务器遇到了一个未曾预料的状态，导致了它无法完成对请求的处理。一般来说，这个问题都会在服务器的程序码出错时出现。
- 501 服务器不支持当前请求所需要的某个功能。当服务器无法识别请求的方法，并且无法支持其对任何资源的请求。
- 502 作为网关或者代理工作的服务器尝试执行请求时，从上游服务器接收到无效的响应。
- 503 由于临时的服务器维护或者过载，服务器当前无法处理请求。这个状况是临时的，并且将在一段时间以后恢复。

6. cookie 和 session 机制的区别

- cookies 数据保存在客户端，session 数据保存在服务器端；
- cookies 可以减轻服务器压力，但是不安全，容易进行 cookies 欺骗；
- session 较安全，但占用服务器资源

7. TCP 和 UDP 的区别

- TCP：面向连接，可靠的，速度慢，效率低
- UDP：无连接、不可靠、速度快、效率高

8. TCP 为什么是三次握手和四次挥手

- 三次握手能保证数据可靠传输又能提高传输效率。若握手是两次：如果只是两次握手，至多只有连接发起方的起始序列号能被确认，另一方选择的序列号则得不到确认；
- 要保证双方都关闭了连接。因为 TCP 是全双工的，就是要等到两边都发送 fin 包确认双方都没有数据传输后才关闭；

9. TCP为什么最后挥手后会有time_wait

- 为了保证可靠的断开TCP的双向连接，确保足够的时间让对方收到 ACK 包。若客户端回复的 ACK 丢失，server 会在超时时间到来时，重传最后一个 fin 包，处于 TIME_WAIT 状态的 client 可以继续回复 Fin 包，发送 ACK。

- 保证让迟来的 TCP 报文段有足够的时间被识别和丢弃，避免新旧连接混淆。有些路由器会缓存没有收到的数据包，如果新的连接开启，这些数据包可能就会和新的连接中的数据包混在一起。连接结束了，网络中的延迟报文也应该被丢弃掉，以免影响立刻建立的新连接。

10. 简要说明 HTTP 请求中的 Post 和 Get 有哪些区别的地方

- 请求头多了 content-length 和 content-type 字段
- Post 可以附加 body，可以支持 form、json、xml、binary 等各种数据格式
- 行业通用规范
- 无状态变化的建议使用 Get
- 数据的写入与状态的修改建议使用 Post
- 基于 HTTP 协议：都是请求返回数据，Get 将请求体放在头上，只发一次请求，Post 将请求体放在内部，需要发送两次请求
- GET 在浏览器回退时是无害的，而 POST 会再次提交请求。
- GET 请求会被浏览器主动 cache，而 POST 不会，除非手动设置。
- GET 请求只能进行 URL 编码，而 POST 支持多种编码方式。
- GET 请求在 URL 中传送的参数是有长度限制的，而 POST 么有。
- 对参数的数据类型，GET 只接受 ASCII 字符，而 POST 没有限制。
- GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。

11. 如果一个请求，返回的状态码是 200，但是没有内容，可能发生了什么？

- 请求头缺失或错误
- 参数 length 不符

Linux 技能篇

1. 工作中常用的 Linux 命令有哪些？`awk、sed、vim、iotop、dstat、cp、top、ifconfig、pwd、cd、ll、ls、cat、tail、grep、mv、rm、mkdir、df、du`

2. 什么命令可以帮助 Linux 执行 Windows 上传的脚本？

- 改变编码格式
- vim test.sh
- :set ff?// 显示dos的话
- :set ff=unix:wq

3. 简述 Linux 三剑客

- grep 命令
 - 根据用户指定的模式 pattern 对目标文本进行过滤，显示被模式匹配到的行；
 - `grep [options] pattern [file]`
 - 常用参数：
 - -v 显示不被pattern匹配到的行
 - -i 忽略字符的大小写
 - -n 显示匹配的行号

- -c 统计匹配的行数
- -o 仅显示匹配到的字符串
- -E 使用ERE，相当于egrep（可以识别更多的正则表达式规则）

o sed 命令

- 流编辑器，用来处理一行数据。将一行数据存储在模式空间中->用sed命令处理->送入屏幕->清空空间。
- 常用参数：
 - -h 显示帮助
 - -n 仅显示script处理后的结果
 - -e 指定的脚本来处理输入的文本文件
 - -f 以指定的脚本文件来处理
- 常用动作：
 - a: 新增 sed -e '4 a newline'
 - c: 取代 sed -e '2,5c No 2-5 number'
 - d: 删除 sed -e '2,5d'
 - i: 插入 sed -e '2i newline'
 - p: 打印 sed -n '/root/p'
 - s: 取代 sed -e 's/old/new/g'
 - g: 代表全局

o awk 命令

-
- 把文件逐行的读入，以空格为默认分隔符将每行切片。把行作为输入，并赋值给\$0->将行切段，从\$1开始->对行匹配正则/执行动作->打印内容；
- `awk 'pattern + action' [filenames]`
- 常用语法：
 - filename awk 浏览的文件名
 - begin 处理文本前要执行的操作
 - end 处理文本之后要执行的操作
 - fs 设置输入域分隔符，等价于命令行-F选项
 - nf 浏览记录的域的个数（列数）
 - nr 已读的记录数（行数）
- 常用参数：
 - ofs 输出域分隔符
 - ors 输出记录分隔符
 - rs 控制记录分隔符，换行标志
 - \$0 整条记录
 - \$1 第一条分隔后的记录

4. 如何通命令定位 Linux 服务器下的日志？

- 如果要监控日志，那么使用 `tail -f | grep xxx` 命令，过滤需要的字段；
- 如果在完整日志中查看内容，使用 `cat xxx.log | grep xxxx | awk '{print $1}'` 等命令过滤自己需要的内容；

5. 简述项目中的环境搭建和维护

- 结合自身经验先从系统安装开始，如常用的 CentOS 和 Ubuntu 说起，系统安装主要是磁盘分区和磁盘阵列问题；
- 基础环境依赖，如 MySQL、Redis、Jenkins、Docker、项目中用到的其他依赖环境等；
- 维护方便主要从遇到的错误说起，如无法远程连接、服务器加固等；

自动化测试相关

包含 Selenium、Appium 和接口测试。

1. 自动化代码中，用到了哪些设计模式？

- 单例模式
- 工厂模式
- PO模式
- 数据驱动模式

2. 什么是断言？

- 检查一个条件，如果它为真，就不做任何事，用例通过。如果它为假，则会抛出 `AssertError` 并且包含错误信息。

3. UI 自动化测试中，如何做集群？

- Selenium Grid，分布式执行用例
- Appium 使用 STF 管理多设备
- Docker+K8S 管理集群

4. 怎么对含有验证码的功能进行自动化测试？

- 万能验证码
- 测试环境屏蔽验证
- 其他操作不推荐

5. 如何优化和提高 Selenium 脚本的执行速度？

- 尽量使用 `by_css_selector()` 方法
- `by_css_selector()` 方法的执行速度比 `by_id()` 方法的更快，因为源码中 `by_id()` 方法会被自动转成 `by_css_selector()` 方法处理；
- 使用等待时，尽量使用显示等待，少用 `sleep()`，尽量不用隐式等待；
- 尽量减少不必要的操作：可以直接访问页面的，不要通过点击操作访问；
- 并发执行测试用例：同时执行多条测试用例，降低用例间的耦合；
- 有些页面加载时间长，可以中断加载；

6. 接口测试能发现哪些问题？

- 可以发现很多在页面上操作发现不了的 bug；
- 检查系统的异常处理能力；
- 检查系统的安全性、稳定性；
- 前端随便变，接口测好了，后端不用变；
- 可以测试并发情况，一个账号，同时（大于 2 个请求）对最后一个商品下单，或不同账号，对最后一个商品下单；
- 可以修改请求参数，突破前端页面输入限制（如金额）；

7. Selenium 中隐藏元素如何定位？

- 如果单纯的定位的话，隐藏元素和普通不隐藏元素定位没啥区别，用正常定位方法就行了（这个很多面试官也搞不清楚）；
- 元素的属性隐藏和显示，主要是 `type="hidden"` 和 `style="display: none;"` 属性来控制的，接下来在元素属性里面让它隐藏，隐藏元素可以正常定位到，只是不能操作（定位元素和操作元素是两码事，很多初学者傻傻分不清楚），操作元素是 `click,clear,send_keys` 这些方法；
- JS 操作隐藏元素；

8. 如何判断一个页面上元素是否存在？

- 方法一：用 `try...except...`
- 方法二：用 `elements` 定义一组元素方法，判断元素是否存在,存在返回 `True`,不存返回 `False`
- 方法三：结合 `WebDriverWait` 和 `expected_conditions` 判断（推荐）

9. 如何提高脚本的稳定性？

- 不要右键复制 `xpath`(十万八千里那种路径，肯定不稳定)，自己写相对路径，多用 `id` 为节点查找；
- 定位没问题，第二个影响因素那就是等待了，`sleep` 等待尽量少用（影响执行时间）；
- 定位元素方法重新封装，结合 `WebDriverWait` 和 `expected_conditions` 判断元素方法，自己封装一套定位元素方法；

10. 如何定位动态元素？

- 动态元素有 2 种情况，一个是属性动态，比如 `id` 是动态的，定位时候，那就不要用 `id` 定位就是了；
- 还有一种情况动态的，那就是这个元素一会在页面上方，一会在下方，飘忽不定的动态元素，定位方法也是一样，按 `f12`，根据元素属性定位（元素的 `tag`、`name` 的步伐属性是不会变的，动的只是 `class` 属性和 `styles` 属性）；

11. 如何通过子元素定位父元素

- 使用 `element.parent` 方法

12. 平常遇到过哪些问题？如何解决的

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

13. 一个元素明明定位到了，点击无效（也没报错），如果解决？

- 使用 JS 点击，Selenium 有时候点击元素是会失效；

14. 测试的数据你放在哪？

- 对于账号密码，这种管全局的参数，可以用命令行参数，单独抽出来，写的配置文件里（如 ini）；
- 对于一些一次性消耗的数据，比如注册，每次注册不一样的数，可以用随机函数生成；
- 对于一个接口有多组测试的参数，可以参数化，数据放 YAML, Text, JSON, Excel 都可以；
- 对于可以反复使用的数据，比如订单的各种状态需要造数据的情况，可以放到数据库，每次数据初始化，用完后再清理；
- 对于邮箱配置的一些参数，可以用 ini 配置文件；
- 对于全部是独立的接口项目，可以用数据驱动方式，用 excel/csv 管理测试的接口数据；
- 对于少量的静态数据，比如一个接口的测试数据，也就 2-3 组，可以写到 py 脚本的开头，十年八年都不会变更的；

15. 什么是数据驱动，如何参数化？

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

16. 其他接口都需要登录接口的信息，怎么去让这个登录的接口只在其他接口调用一次？

- 使用单例模式
- 使用自定义缓存机制
- 使用测试框架中的 setup 机制
- pytest 中 fixture 机制

17. 接口产生的垃圾数据如何清理？

- 造数据和数据清理，需用 python 连数据库了，做增删改查的操作测试用例前置操作，setUp 做数据准备后置操作，tearDown 做数据清理

18. 怎么用接口案例去覆盖业务逻辑？

- 考虑不同的业务场景，一个接口走过的流程是什么样的，流程的逻辑是什么样的，什么样的参数会有什么样的结果，多场景覆盖；

性能篇

1. 性能测试指标包括哪些

- 最大并发用户数，HPS（点击率）、事务响应时间、每秒事务数、每秒点击量、吞吐量、CPU 使用率、物理内存使用、网络流量使用等。
- 前端需主要关注的点是：
 - 响应时间：用户从客户端发出请求，并得到响应，以及展示出来的整个过程的时间。
 - 加载速度：通俗的理解为页面内容显示的快慢。
 - 流量：所消耗的网络流量。
- 后端需主要关注的是：
 - 响应时间：接口从请求到响应、返回的时间。

- 并发用户数：同一时间点请求服务器的用户数，支持的最大并发数。
- 内存占用：也就是内存开销。
- 吞吐量（TPS）：Transaction Per Second, 每秒事务数。在没有遇到性能瓶颈时：TPS=并发用户数*事务数/响应时间。
- 错误率：失败的事务数/事务总数。
- 资源使用率：CPU占用率、内存使用率、磁盘I/O、网络I/O。
- 从性能测试分析度量的角度来看，主要可以从如下几个大的维度来收集考察性能指标：
- 系统性能指标、资源性能指标、稳定性指标

2. 如果一个需求没有明确的性能指标，要如何开始进行性能测试？

- 先输出业务数据，如 pv、pu、时间段等，计算出大概的值，然后不断加压测到峰值

3. 介绍 JMeter 聚合报告包括哪些内容？

- 请求名、线程数、响应时间（50 95 99 最小 最大）错误率、吞吐量

4. 如果有一个页面特别卡顿，设想一下可能的原因？

- 后台：接口返回数据慢，查询性能等各种问题
- 前端：使用 Chrome 工具调试，判断 JS 执行久或是其他问题
- 网络问题

5. 说一说项目中的实际测试内容

- 根据自己项目中的经验实话实说，有没有经验很容易露馅。

6. 介绍一下 JMeter 进行性能测试的过程

- 结合自己的项目经验聊。大家也可以自行搜索。

7. 介绍一下 JMeter 和 LoadRunner 的区别

- 详细的不展开了，最重要的是相对来说 LoadRunner 的笨重、昂贵、闭源，理念和生态都落后，而 JMeter 是开源、可定制化开发，功能强大易用，并且在互联网大厂都已经非常有成熟的落地方案（主流的互联网公司基本都在使用 JMeter+ELK+Grafana+Influxdb 这套架构），可以说是进 BAT 大厂必备技能。还不会 JMeter 的同学建议抓紧补起来。

软素质篇（10 大灵魂拷问）

1. 说一下自己的优点和缺点？

- 避开岗位的核心技能
- 把缺点放在场景中描述
- 对缺点进行合理化解释
- 优点随便说，主要方向还是在岗位上

2. 是否能接受加班？（建议分情况作答）

- 第一种情况：假设公司有重要的项目要赶。示范回答：贵公司现在正处于发展上升期，也在官网上有看到公司的重要项目成果，我觉得有时候因为赶项目进度、工作需要等忙起来是非常正

常的，面对这种情况，我是非常愿意配合公司和团队的工作，让工作能够更顺利地完成，此外，我也相信自己一定能在公司安排的工作中获得锻炼，获得更快地成长。

- 第二种情况：假设自己作为新人，对业务不熟悉。示范回答：我作为公司刚进去的新人，可能刚开始进入公司接触业务时不太熟练，会出现需要加班的情况，但我更愿意提高工作效率，并积极向公司的前辈请教学习，在一定的时间内完成工作而不是拖到下班之后。当然，如果有紧急的事情，忙起来需要加班也是可以接受的。

3. 你对薪酬的要求？

- 我上家公司基本在A~B 之间（建议合理提高，避免部分HR压价）。
- 薪资并不是我求职的唯一标准，我来贵司求职的主要动机是兴趣，这份工作是我喜欢做的，也相信自己可以胜任，更相信公司会给出一个合理的薪酬。
- 相比薪酬，我更在意的是收入，所以，我很愿意了解贵司的薪酬架构，可以简单介绍下吗？
- 我希望薪资可以达到 XX，据我了解，贵司这个岗位薪资范围是A~B，而结合岗位职责及任职要求，我对自己也进行了相应评估，也愿意接受贵司的下一步考核。

4. 未来 5 年的职业规划

- 自我认知。对自己是否了解，了解是不是靠谱。
- 动机和价值观。你是否能接受我们并不一定能给你公平的职业发展机会这个现实？
- 组织承诺。你到底能在我们这踏实的干几年？

5. 为什么你觉得这个岗位适合自己？（为什么要聘用你）

- 描述应聘岗位的胜任条件，强调自己的工作能力和工作经验跟岗位的匹配度，岗位要求的工作技能是否自己掌握了，掌握的程度是怎样的，最好在面试中说出来。因此，在面试前最好是要针对应聘岗位，把自己胜任的条件——列出来，做到知己知彼。可以谈论一下自己之前的工作情况，用成绩、用数据来说明自己的成就。
- 描述自己能为公司做出什么贡献，公司是一个讲究利益的地方，聘用你肯定要你为公司做出贡献。那么你在回答这个问题时，就需要说出你的加入可以为公司带来什么，这非常重要。因此，一定要明确你的工作目标和职业规划，表明你的立场和专业程度，让HR信任你。
- 描述出自身的优势。公司为何要聘用你，而不聘用别人，肯定是有比别人优秀的地方。那么在回答这个问题时，就一定要说出自己与众不同的地方，最好是要举一个例子，来支持你的观点。
- 建立个人和公司的联系，HR想要得到一个怎样的答案呢？无非就是想通过这个问题，来进一步了解你各方面的信息，以及看看你为这次面试做了多少功课。那么在面试前，你最好是要尽可能获取有关公司可行业的资料信息。在回答的时候，结合自己所做的功课，建立个人和公司的联系，说明自己在哪一方面能够匹配公司的要求。HR看到你对应聘岗位这么了解，肯定会对你有好感。
- 说出你对这份工作的兴趣以及热情。

6. 对我们公司有多少了解？

- 如果不了解，就按实际情况回答就好，知道多少就说多少，（很多时候去面试对这个公司的了解都是从网上查到的，不会太深入）；

- 但最好是提前做好一些调研和准备工作；

7. 为什么愿意到我们公司？

- 有所准备，了解公司基本情况
- 个人目标与公司目标一致
- 强调你能如何为公司创造价值

8. 与领导 / 团队同事意见不一致时，该如何处理？

- 不要假设“我已经完全的掌握了对这件事的认知”。向领导询问确认自己有可能缺失的信息。要寻找对领导没有告知的信息，和领导不能透露的信息。
- 不要假设领导已经完全的掌握了我对这件事的认知。检查一下，是否已经将事情的前因后果，自己对事情的理解，明确清晰的传达给了领导，以及，他是否真的已经明确了解。
- 在进行有效的认知沟通后，重新思考整件事情。如果意见还是有不一致，那么：
 - 按领导要求执行。不理解，也执行，在执行中理解。
 - 执行过程中，收集反馈，不断调整，提升认知。
 - 执行完成后，及时复盘，回顾决策和行动过程，沉淀知识。

9. 缺乏工作经验，如何胜任这份工作？

- 承认工作经验的重要性。这样能带给面试官的印象是：该位候选人认知能力较强，具有理性思维与客观公正的处事态度及判断能力，尤其是对于自己也能客观公正地看待，勇于承认自己的缺失。
- 突显个人优势。用自己的其他优势特长来补足经验上的不足，比如说记忆力好、动手能力强、语言能力强、学习能力强等。
- 强调自己会不断提高工作能力。切忌用假大空的话，要用具体的与工作相关的事例或是数据来说明自己的学习力。

10. 工作 / 会议中与同事发生争执，如何处理？

- 在沟通之前，做好充分的准备
- 学会认真倾听，让别人把话说完
- 发现有情绪化苗头时，及时停止会议
- 借用一些工具，来解决交流障碍
- 学会非暴力沟通的方式

反问面试官

1. 职责

- 团队中初级和高级人员如何平衡
- 针对员工有哪些培训和提升计划

2. 技术

- 公司内部的技术栈
- 产品的架构
- 版本控制及迭代速度

- 服务器管理权限，本地计算机管理权限

3. 团队

- 团队内和团队之间如何沟通
- 遇到了分歧如何解决
- 团队正在经历的尚未解决的挑战是什么
- 绩效考核是如何算的

4. 公司

- 晋升机会
- 是否有自己的学习资源
- 假期，加班工资等
- 过去半年最糟糕的一天是怎么样的
- 是什么让你来到并留在这里
- 是否能够平衡工作与生活