自动化基础篇

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

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。

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

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

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；

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）；

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 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；

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）

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；

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）；

如何通过子元素定位父元素

使用element.parent方法

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；

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；

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；

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；

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

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

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

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度角来看，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

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

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

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

说一说项目中的实际测试内容

首先介绍项目背景，然后介绍项目的流量，我们这个项目目前日活(每天活跃用户数)大约在10w左右，平均在线时长十分钟左右，QPS峰值600，谷值500，TPS在100左右，基本上量说大也不大，老板应该还在谈A轮的融资，当然了，为了拉投资，有一些量应该是内部刷出来的。

项目迭代/开发周期：我们公司一般是四周左右一个大版本，两周左右一个小版本进行迭代，也就是一个月升级一个大版本，半个月升级一个小版本，测试周期占开发周期的百分之三十左右，当然了，有时候也会有发小版本做紧急修复或者特殊功能的更新。

这个项目我主要参与的模块有：用户模块、积分模块、检索模块

首先用户模块包含，用户的一些基本操作，比如注册/登录，这里在注册时候，除了一些基本的参数合法性验证，我还针对mysql的唯一索引进行了验证性测试，也就是用户名不能重复，这里用mysql唯一索引进行限制操作

用户登录的时候，令我印象非常深刻的一次是进行接口安全性测试，这次安全性测试主要是为了检测接口是否有防御sql注入攻击的能力，我们知道,mysql注释符号是#，所以在我负责测试的登录模块里，我

针对登录接口进行sql注入攻击，正常情况下，接口应该返回非法用户，而不是登录成功，结果接口返回登录成功，这里我说一下注入攻击的原理，通过#将用户得密码进行屏蔽操作，这样，在后台接口中sql语句就只有where username = 而不会将and password = 加入sql语句中，导致无需密码既可以登录。

积分模块，每当有用户发起消费积分的请求时，积分的逻辑就是递减，当用户请求非常少的时候，这个逻辑完全没有问题，测试用例全部通过，但是当我使用apachebench做高并发压力测试的时候，用户积分就会突然然变成负数，这个bug的严重程度非常致命，如果贸然部署生产环境，就会法生灾难性的事故，比如说拼多多事件，经过和RD的联合调试，发现原因是由于积分递减逻辑没有进行加锁控制，当当量的消费请求打过来，递减的同时无法判断积分余额，所以导致了bug的出现

检索模块，检索是每一个应用都具备通用模块，也是一个应用中经常被用到的高频模块，一般高频操作的模块，一定得进行性能测试，当时我用的Jmeter，我们这个应用目前日活，每日活跃用户数在8W左右，那么接口需要满足的并发压力就在500左右，此时，我用jemter模拟总量800请求，每秒600请求，进行批量操作，此时我发现服务器cpu的占用量已经超过80%,所以发现了这个问题以后，我立刻将情况汇报给RD，责令其进行接口优化或者服务器扩容。

检索模块还有一个重要功能，就是分页，我们项目后台分页逻辑是根据mysql关键字limit来决定，limit原理，第一个参数为获取下标位置，第二个参数是获取分页长度，那么根据分页原理，开始位置=（当前页-1） * 页码，结束位置当前页* 页码，所以我根据这套算法，对分页接口进行功能测试，印证数据一致性，同时海量数据情况下，比如500万用户量，每页展示50个，此时页数巨大，我根据抽样算法以及边界值推算进行验证。

介绍一下 JMeter 和 LoadRunner 的区别

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有非常成熟的落地方案（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。