**一、你以前工作的测试流程是什么？**

**答：公司对测试流程没有规定如何做，但每个测试人员都有自己的一套测试流程。我说下我 1 年来不断改正 （自己总结，吸取同行的方法）后的流程吧。需求评审（有开发人员，产品经理，测试人员，项目经理）－> 需求确定(出一份确定的需求文档)－>开发设计文档（开发人员在开始写代码前就能输出设计文档）－>想好测 试策略，写出测试用例－>发给开发人员和测试经理看看（非正式的评审用例）－>接到测试版本－>执行测试 用例（中间可能会补充用例）－>提交 bug（有些 bug 需要开发人员的确定（严重级别的，或突然发现的在测 试用例范围之外的，难以重现的），有些可以直接录制进 TD）－>开发人员修改（可以在测试过程中快速的修 改）－>回归测试（可能又会发现新问题，再按流程开始跑）。**

**描述软件产生内存泄露的原因以及检查方式。（可以结合一种开发语言进行描述）**

**内存泄露的原因，主要是由于开发过程当中申请了计算机资源（例如对象、内存等），但是使用资源完成 以后没有及时释放资源导致的。例如在 C 语言当中使用了 malloc 申请了内存，但是未使用 free 来释放内存。**

**二、简述什么是值传递，什么是地址传递，两者区别是什么？**

**答：值传递主调函数传递给被调函数的是值的拷贝，不是原值；地址传递主调函数传递给被调函数的是值的地 址。区别是值传递被调函数中的操作不改变主调函数的值，而地址传递则不同。**

**三、结构化程序设计和面向对象程序设计各自的特点及优缺点是什么？**

**答：（不需要回答如此复杂） 结构化程序设计思想采用了模块分解与功能抽象和自顶向下、分而治之的方法，从而有效地将一个较复杂 的程序系统设计任务分解成许多易于控制和处理的子程序，便于开发和维护。它的重点在于把功能进行分解。 但是由于在实际开发过程当中需求会经常发生变化，因此，它不能很好的适应需求变化的开发过程。结构化程 序设计是面向过程的。 面向对象程序设计以需求当中的数据作为中心，来进行设计，具有良好的代码重用性。 封装性：也叫数据隐藏，用户无需知道内部工作流程，只要知道接口和操作就可以的，C++中一般用类来实现 封装。继承性： 一种支持重用的思想，在现有的类型派生出新的子类，例如新型电视机在原有型号的电视机上增 加若干中功能而得到，新型电视机是原有电视机的派生，继承了原有电视机的属性，并增加了新的功能。 多态性：指在一般类中定义的属性或行为，被特殊类继承之后，可以具有不同的数据类型或表现出不同的 行为。动态联编：指一个计算机程序自身彼此关联的过程，按照联编所进行的阶段不同，可分为两种不同的联编 方法：静态联编和动态联编**

**四、简述什么是存储过程和触发器？**

**答： 存储过程：是数据库中的一个对象，Transact-SQL 语句的预编译集合，这些语句在一个名称下存储并作为 一个单元进行处理。（可以理解为 C 语言中的函数，有参数、返回值等函数特性） 触发器是一种特殊类型的存储过程，当使用下面的一种或多种数据修改操作在指定表中对数据进行修改 时，触发器会生效：UPDATE、INSERT 或 DELETE。**

**五、使用 C 语言编写一个函数，用于交换两个变量的值（地址传递）。**

**答：**

**void Swap(int \*a,int \*b)**

**{**

**int temp;**

**int temp=\*a;**

**int \*a=\*b; int \*b=temp;**

**}**

**1**

**2**

**3**

**4**

**5**

**6**

**六、请简述 DNS、活动目录、域的概念。**

**答： DNS：域名服务，作用是将网络域名解析成 IP 地址； 活动目录：微软提供的目录服务的一种，它存储有关网络上的对象信息，并使管理员和用户更方便的查找和使 用这类信息； 域：网络系统的一个安全边界，在一个域当中，计算机和用户共享一些列的安全信息。**

**七、在 RedHat 中，从 root 用户切到 userl 用户，一般用什么命令？**

**答：su su user1 切换到 user1，但切换后的当前目录还是 root 访问的目录 su – user1 切换到 user1，并且当前目录切换到 user1 的根目录下（/home/user1/）**

**八、你的测试职业发展是什么？你自认为做测试的优势在哪里？**

**答： 测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级测试工程师奔 去。而且我也有初步的职业规划，前 3 年累积测试经验，按如何做好测试工程师的要求自己，不断的更新自己改正自 己 ， 做好测试任务 。 优势在于我对测试坚定不移的信心和热情，虽然经验还不够，但测试需要的基本技能我有信心在工作中得以发挥。**

**九、测试产品与测试项目的区别是什么？**

**答： 习惯上把开发完成后进行商业化、几乎不进行代码修改就可以售给用户使用的软件成为软件产品，也 就是可以买“卖拷贝”的软件，例如 Windows2000。而通常把针对一个或者几个特定的用户而开发的软件 成为软件项目，软件项目是一种个性化的产品，可以是按照用户要求全部重新开发，也可以修改已有的软 件产品来满足特定的用户需求。项目和产品的不同特点，决定我们测试产品和测试项目仍然会有很多不同 的地方：**

**质量要求不同。通常产品的质量要高一些，修复发布后产品的缺陷成本较高，甚至会带来很多负面 的影响。而做项目通常面向某一用户，虽然质量越高越好，但是一般只要满足用户要求就可以了。**

**测试资源投入多少不同。做软件产品通常是研发中心来开发，进度压力要小些。同时由于质量要求 高，因此会投入较多的人力、物力资源。**

**项目最后要和用户共同验收测试，这是产品测试不具有的特点。 此外，测试产品与测试项目在缺陷管理方面、测试策略制定都会有很大不同，测试管理者应该结合具 体的环境，恰如其分的完成工作。**

**十、测试工具在测试工作中是什么地位？**

**答： 国内的很多测试工程师对测试工具相当迷恋，尤其是一些新手，甚至期望测试工具可以取代手工测 试。测试工具在测试工作中起的是辅助作用，一般用来提高测试效率。自动化测试弥补了手工测试的不 足，减轻一定的工作量。实际上测试工具是无法替代大多数手工测试的，而一些诸如性能测试等自动化测 试也是手工所不能完成的。 对于自动测试技术，应当依据软件的不同情况来分别对待，一般自动技术会应用在引起大量重复性工 作的地方、系统的压力点、以及任何适合使用程序解决大批量输入数据的地方。然后再寻找合适的自动测 试工具，或者自己开发测试程序。一定不要为了使用测试工具而使用。**

**————————————————**

**版权声明：本文为CSDN博主「测试人虚竹」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。**

**原文链接：<https://blog.csdn.net/qq_40214204/article/details/108836040>**

2021最详细最全的软件测试面试题，吊打面试官，大把offer手到擒来！

软件测试小dao 2021-05-11 16:02:36 430 收藏 5

分类专栏： 软件测试 测试工程师 面试 文章标签： 测试工程师 软件测试 面试 offer 工作

版权

求职，类似于打仗，是一场挑战自己的战斗，也是一场跟用人单位的博弈，更是一场千人过独木桥的厮杀、混战。《孙子·谋攻篇》中早就说了：“知己知彼，百战不殆；不知彼而知己，一胜一负；不知彼，不知己，每战必殆。”。

在当今竞争激烈的软件测试职场中，同学们想谋一份令他人羡慕、让自己欣喜若狂的工作，事先充分准备是十分必要的。若能在笔试、面试中不断认识并提高自己，不断了解用人单位和面试对手、判官，不断改进进攻对策，那么离理想的职位也就不远了。

在这里呢，小dao专门给各位即将踏入面试征程的软件测试小伙伴们准备了一份最全的软件测试面试问题和答案，通读全文并背熟后，保证你驰骋测试面试场，大把offer手到擒来！不要太感动喔。

1：测试中发现了一个bug，但是开发经理认为这不是一个bug，你应该怎样解决?

答：首先呢将问题提交到缺陷管理库里面进行备案。

然后就要获取判断的依据和标准：

根据需求说明书、产品说明、设计文档等，确认实际结果是否与计划有不一致的地方，提供缺陷是否确认的直接依据;

如果没有文档依据，可以根据类似软件的一般特性来说明是否存在不一致的地方，来确认是否是缺陷;

根据用户的一般使用习惯，来确认是否是缺陷;

与设计人员、开发人员和客户代表等相关人员探讨，确认是否是缺陷;

合理的论述，向测试经理说明自己的判断的理由，注意客观、严谨，不参杂个人情绪。

等待测试经理做出最终决定，如果仍然存在争议，可以通过公司政策所提供的渠道，向上级反映，并有上级做出决定。

2：给你一个网站，你如何测试?

答.首先，查找需求说明、网站设计等相关文档，分析测试需求。

制定测试计划，确定测试范围和测试策略，一般包括以下几个部分：功能性测试;界面测试;性能测试;数据库测试;安全性测试;兼容性测试

设计测试用例：

功能性测试可以包括，但不限于以下几个方面：

链接测试。链接是否正确跳转，是否存在空页面和无效页面，是否有不正确的出错信息返回。

提交功能的测试。

多媒体元素是否可以正确加载和显示。

多语言支持是否能够正确显示选择的语言等。

界面测试可以包括但不限于一下几个方面：

页面是否风格统一，美观

页面布局是否合理，重点内容和热点内容是否突出

控件是否正常使用

.对于必须但未安装的控件，是否提供自动下载并安装的功能

文字检查

性能测试一般从以下两个方面考虑：

压力测试;负载测试;强度测试

数据库测试要具体决定是否需要开展。数据库一般需要考虑连结性，对数据的存取操作，数据内容的验证等方面。

安全性测试：

基本的登录功能的检查

是否存在溢出错误，导致系统崩溃或者权限泄露

相关开发语言的常见安全性问题检查，例如SQL注入等

如果需要高级的安全性测试，确定获得专业安全公司的帮助，外包测试，或者获取支持

兼容性测试，根据需求说明的内容，确定支持的平台组合：

浏览器的兼容性;

操作系统的兼容性;

软件平台的兼容性;

数据库的兼容性

开展测试，并记录缺陷。合理的安排调整测试进度，提前获取测试所需的资源，建立管理体系(例如，需求变更、风险、配置、测试文档、缺陷报告、人力资源等内容)。

定期评审，对测试进行评估和总结，调整测试的内容。【推荐了解：软件测试课程】

3：在搜索引擎中输入汉字就可以解析到对应的域名，请问如何用LoadRunner进行测试。

答：建立测试计划，确定测试标准和测试范围

设计典型场景的测试用例，覆盖常用业务流程和不常用的业务流程等

根据测试用例，开发自动测试脚本和场景：

录制测试脚本：新建一个脚本(Web/HTML协议);点击录制按钮，在弹出的对话框的URL中输入”about:blank”;在打开的浏览器中进行正常操作流程后，结束录制;调试脚本并保存，可能要注意到字符集的关联。

设置测试场景：针对性能设置测试场景，主要判断在正常情况下，系统的平均事务响应时间是否达标;针对压力负载设置测试场景，主要判断在长时间处于满负荷或者超出系统承载能力的条件下，系统是否会崩溃;执行测试，获取测试结果，分析测试结果

4：一台客户端有三百个客户与三百个客户端有三百个客户对服务器施压，有什么区别?

答：　300个用户在一个客户端上，会占用客户机更多的资源，而影响测试的结果。线程之间可能发生干扰，而产生一些异常。

300个用户在一个客户端上，需要更大的带宽。

IP地址的问题，可能需要使用IP Spoof来绕过服务器对于单一IP地址最大连接数的限制。

所有用户在一个客户端上，不必考虑分布式管理的问题;而用户分布在不同的客户端上，需要考虑使用控制器来整体调配不同客户机上的用户。同时，还需要给予相应的权限配置和防火墙设置。

5：试述软件的概念和特点?软件复用的含义?构件包括哪些?

答：　软件是计算机系统中与硬件相互依存的另一部分，与计算机系统操作有关的计算机程序、规程、规则，以及可能有的文件、文档及数据。

软件复用(SoftWare Reuse)是将已有软件的各种有关知识用于建立新的软件，以缩减软件开发和维护的花费。软件复用是提高软件生产力和质量的一种重要技术。早期的软件复用主要是代码级复用，被复用的知识专指程序，后来扩大到包括领域知识、开发经验、设计决定、体系结构、需求、设计、代码和文档等一切有关方面。

可以被复用的软件成分一般称作可复用构件。

6：软件生存周期及其模型是什么?

答：　软件生存周期(Software life cycle)又称为软件生命期，生存期。是指从形成开发软件概念起，所开发的软件使用以后，知道失去使用价值消亡为止的整个过程。一般来说，整个生存周期包括计划(定义)、开发、运行(维护)三个时期，每个时期又划分为若干个阶段。每个阶段有明确的任务。

周期模型(典型的几种)：

瀑布模型

快速原型模型：快速原型模型允许在需求分析阶段对软件的需求进行初步而非完全的分析和定义，快速设计开发出软件系统的原型，该原型向用户展示待开发软件的全部或部分功能和性能;用户对该原型进行测试评定，给出具体改进意见以丰富细化软件需求;开发人员据此对软件进行修改完善，直至用户满意认可之后，进行软件的完整实现及测试、维护。

迭代模型：迭代包括产生产品发布(稳定、可执行的产品版本)的全部开发活动和要使用该发布必需的所有其他外围元素。在某种程度上，开发迭代是一次 完整地经过所有工作流程的过程：需求分析、设计、实施和测试工作流程。实质上，它类似小型的瀑布式项目。RUP认为，所有的阶段都可以细分为迭代。每一次 的迭代都会产生一个可以发布的产品，这个产品是最终产品的一个子集。

生命周期阶段：

软件计划与可行性分析

需求分析

软件设计

编码

软件测试

运行与维护

7：什么是软件测试?软件测试的目的与原则

答：在规定的条件下对程序进行操作，以发现程序错误，衡量软件质量，并对其是否能满足设计要求进行评估的过程。

软件测试的目的：

测试是程序的执行过程，目的在于发现错误

一个成功的测试用例在于发现至今未发现的错误

一个成功的测试是发现了至今未发现的错误的测试

确保产品完成了它所承诺或公布的功能，并且用户可以访问到的功能都有明确的书面说明。

确保产品满足性能和效率的要求

确保产品是健壮的和适应用户环境的

软件测试的原则：

测试用例中一个必须部分是对预期输出或接过进行定义

程序员应避免测试自己编写的程序

编写软件的组织不应当测试自己编写的软件

应当彻底检查每个测试的执行结果

测试用例的编写不仅应当根据有效和预料到的输入情况，而且也应当根据无效和未预料到的输入情况

检擦程序是否“未做其应该做的”仅是测试的一半，测试的另一半是检查程序是否“做了其不应该做的”

应避免测试用例用后即弃，除非软件本身就是个一次性的软件

计划测试工作时不应默许假定不会发现错误

程序某部分存在更多错误的可能性，与该部分已经发现错误的数量成正比

软件测试是一项极富创造性，极具智力的挑战性的工作

8：：软件配置管理的作用?软件配置包括什么?

答：软件配置管理(Software Configuration Management，SCM)是一种标识、组织和控制修改的技术。软件配置管理应用于整个软件工程过程。在软件建立时变更是不可避免的，而变更加剧了项目中软件开发者之间的混乱。SCM活动的目标就是为了标识变更、控制变更、确保变更正确实现并向其他有关人员报告变更。从某种角度讲，SCM是一种标识、组织和控制修改的技术，目的是使错误降为最小并最有效地提高生产效率。

软件配置包括如下内容：配置项识别、工作空间管理、版本控制、变更控制、状态报告、配置审计

9：什么是软件质量?

答：概括地说，软件质量就是“软件与明确的和隐含的定义的需求相一致的程度”。具体地说，软件质量是软件符合明确叙述的功能和性能需求、文档中明确描述 的开发标准、以及所有专业开发的软件都应具有的隐含特征的程度。 影响软件质量的主要因素，这些因素是从管理角度对软件质量的度量。可划分为三组，分别反应用户在使用软件产品时的三种观点。正确性、健壮性、效率、完整性、可用性、风险(产品运行);可理解性、可维修性、灵活性、可测试性(产品修改);可移植性、可再用性、互运行性(产品转移)。

10：目前主要的测试用例设计方法是什么?

答：白盒测试：逻辑覆盖、循环覆盖、基本路径覆盖

黑盒测试：边界值分析法、等价类划分、错误猜测法、因果图法、状态图法、测试大纲法、随机测试、场景法

11：软件的安全性应从哪几个方面去测试?

答：软件安全性测试包括程序、数据库安全性测试。根据系统安全指标不同测试策略也不同。

用户认证安全的测试要考虑问题： 明确区分系统中不同用户权限 、系统中会不会出现用户冲突 、系统会不会因用户的权限的改变造成混乱 、用户登陆密码是否是可见、可复制 、是否可以通过绝对途径登陆系统(拷贝用户登陆后的链接直接进入系统)、用户退出系统后是否删除了所有鉴权标记，是否可以使用后退键而不通过输入口令进入 系统 、系统网络安全的测试要考虑问题 、测试采取的防护措施是否正确装配好，有关系统的补丁是否打上 、模拟非授权攻击，看防护系统是否坚固 、采用成熟的网络漏洞检查工具检查系统相关漏洞(即用最专业的黑客攻击工具攻击试一下，现在最常用的是 NBSI 系列和 IPhacker IP ) 、采用各种木马检查工具检查系统木马情况 、采用各种防外挂工具检查系统各组程序的外挂漏洞

数据库安全考虑问题： 系统数据是否机密(比如对银行系统，这一点就特别重要，一般的网站就没有太高要求)、系统数据的完整性(我刚刚结束的企业实名核查服务系统中就曾存在数据 的不完整，对于这个系统的功能实现有了障碍) 、系统数据可管理性 、系统数据的独立性 、系统数据可备份和恢复能力(数据备份是否完整，可否恢复，恢复是否可以完整)

12：什么是测试用例 什么是测试脚本 两者的关系是什么?

答：为实施测试而向被测试系统提供的输入数据、操作或各种环境设置以及期望结果的一个特定的集合。

测试脚本是为了进行自动化测试而编写的脚本。

测试脚本的编写必须对应相应的测试用例

13：简述什么是静态测试、动态测试、黑盒测试、白盒测试、α测试 β测试

答：静态测试是不运行程序本身而寻找程序代码中可能存在的错误或评估程序代码的过程。

动态测试是实际运行被测程序，输入相应的测试实例，检查运行结果与预期结果的差异，判定执行结果是否符合要求，从而检验程序的正确性、可靠性和有效性，并分析系统运行效率和健壮性等性能。

黑盒测试一般用来确认软件功能的正确性和可操作性,目的是检测软件的各个功能是否能得以实现,把被测试的程序当作一个黑盒,不考虑其内部结构,在知道该程序的输入和输出之间的关系或程序功能的情况下,依靠软件规格说明书来确定测试用例和推断测试结果的正确性。

白盒测试根据软件内部的逻辑结构分析来进行测试,是基于代码的测试，测试人员通过阅读程序代码或者通过使用开发工具中的单步调试来判断软件的质量，一般黑盒测试由项目经理在程序员开发中来实现。

α测试是由一个用户在开发环境下进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的受控测试，Alpha测试不能由程序员或测试员完成。

β测试是软件的多个用户在一个或多个用户的实际使用环境下进行的测试。开发者通常不在测试现场，Beta测试不能由程序员或测试员完成。

14：软件质量保证体系是什么 国家标准中与质量保证管理相关的几个标准是什么?他们的编号和全称是什么?

答：SQA由一套软件工程过程和方法组成，以保证(软件的)质量。SQA贯穿整个软件开发过程，(它)应包括需求文档评审、代码控制、代码评审、变更管理、配置管理、版本管理和软件测试。

软件质量保证(SQA-Software Quality Assurance)是建立一套有计划，有系统的方法，来向管理层保证拟定出的标准、步骤、实践和方法能够正确地被所有项目所采用。软件质量保证的目的是使软件过程对于管理人员来说是可见的。它通过对软件产品和活动进行评审和审计来验证软件是合乎标准的。软件质量保证组在项目开始时加粗样式就一起参与建立计划、标准和过程。这些将使软件项目满足机构方针的要求。

15：软件产品质量特性是什么?

答：　功能性：适应性、准确性、互操作性、依从性、安全性。

可靠性：成熟性、容错性、易恢复性。

可使用性：易理解性、易学习性、易操作性。

效率：时间特性、资源特性。

可维护性：易分析性、易变更性、稳定性、易测试性。

可移植性： 适应性、易安装性、遵循性、易替换性

16：软件测试的策略是什么?

答：软件测试策略：在一定的软件测试标准、测试规范的指导下，依据测试项目的特定环境约束而规定的软件测试的原则、方式、方法的集合。

17：软件测试分为几个阶段 各阶段的测试策略和要求是什么?

答：和开发过程相对应，测试过程会依次经历单元测试、集成测试、系统测试、验收测试四个主要阶段：

单元测试：单元测试是针对软件设计的最小单位––程序模块甚至代码段进行正确性检验的测试工作，通常由开发人员进行。

集成测试：集成测试是将模块按照设计要求组装起来进行测试，主要目的是发现与接口有关的问题。由于在产品提交到测试部门前，产品开发小组都要进行联合调试，因此在大部分企业中集成测试是由开发人员来完成的。

系统测试：系统测试是在集成测试通过后进行的，目的是充分运行系统，验证各子系统是否都能正常工作并完成设计的要求。它主要由测试部门进行，是测试部门最大最重要的一个测试，对产品的质量有重大的影响。

验收测试：验收测试以需求阶段的《需求规格说明书》为验收标准，测试时要求模拟实际用户的运行环境。对于实际项目可以和客户共同进行，对于产品来说就是最后一次的系统测试。测试内容为对功能模块的全面测试，尤其要进行文档测试。

单元测试测试策略：

自顶向下的单元测试策略：比孤立单元测试的成本高很多，不是单元测试的一个好的选择。

自底向上的单元测试策略：比较合理的单元测试策略，但测试周期较长。

孤立单元测试策略：最好的单元测试策略。

集成测试的测试策略：

大爆炸集成：适应于一个维护型项目或被测试系统较小

自顶向下集成：适应于产品控制结构比较清晰和稳定;高层接口变化较小;底层接口未定义或经常可能被修改;产口控制组件具有较大的技术风险，需要尽早被验证;希望尽早能看到产品的系统功能行为。

加粗样式

　　自底向上集成：适应于底层接口比较稳定;高层接口变化比较频繁;底层组件较早被完成。

基于进度的集成

优点：具有较高的并行度;能够有效缩短项目的开发进度。

缺点：桩和驱动工作量较大;有些接口测试不充分;有些测试重复和浪费。

系统测试的测试策略：

数据和数据库完整性测试;功能测试;用户界面测试;性能评测;负载测试;强度测试;容量测试;安全性和访问控制测试;故障转移和恢复测试;配置测试;安装测试;加密测试;可用性测试;版本验证测试;文档测试

18：软件测试各个阶段通常完成什么工作?各个阶段的结果文件是什么?包括什么内容?

答:　单元测试阶段：各独立单元模块在与系统地其他部分相隔离的情况下进行测试，单元测试针对每一个程序模块进行正确性校验，检查各个程序模块是否正确地实现了规定的功能。生成单元测试报告，提交缺陷报告。

集成测试阶段：集成测试是在单元测试的基础上，测试在将所有的软件单元按照概要设计规格说明的要求组装成模块、子系统或系统的过程中各部分工作是否达到或实现相应技术指标及要求的活动。该阶段生成集成测试报告，提交缺陷报告。

系统测试阶段：将通过确认测试的软件，作为整个给予计算机系统的一个元素，与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合在一起，在实际运行环境下，对计算机系统进行全面的功能覆盖。该阶段需要提交测试总结和缺陷报告。【推荐了解黑马软件测试课程】

19：测试人员在软件开发过程中的任务是什么?

答：1、尽可能早的找出系统中的Bug;

2、避免软件开发过程中缺陷的出现;

3、衡量软件的品质，保证系统的质量;

4、关注用户的需求，并保证系统符合用户需求。

总的目标是：确保软件的质量。

20:在您以往的工作中，一条软件缺陷(或者叫Bug)记录都包含了哪些内容?如何提交高质量的软件缺陷(Bug)记录?

答：一条Bug记录最基本应包含：

bug编号;bug严重级别，优先级;bug产生的模块;首先要有bug摘要，阐述bug大体的内容;bug对应的版本;bug详细现象描述，包括一些截图、录像…等等;bug出现时的测试环境，产生的条件即对应操作步骤;高质量的Bug记录：

通用UI要统一、准确缺陷报告的UI要与测试的软件UI保持一致，便于查找定位。

尽量使用业界惯用的表达术语和表达方法使用业界惯用的表达术语和表达方法，保证表达准确，体现专业化。

每条缺陷报告只包括一个缺陷每条缺陷报告只包括一个缺陷，可以使缺陷修正者迅速定位一个缺陷，集中精力每次只修正一个缺陷。校验者每次只校验一个缺陷是否已经正确修正。

不可重现的缺陷也要报告首先缺陷报告必须展示重现缺陷的能力。不可重现的缺陷要尽力重现，若尽力之后仍不能重现，仍然要报告此缺陷，但在报告中要注明无法再现，缺陷出现的频率。

明确指明缺陷类型根据缺陷的现象，总结判断缺陷的类型。例如，即功能缺陷、界面缺陷、数据缺陷，合理化建议这是最常见的缺陷或缺陷类型，其他形式的缺陷或缺陷也从属于其中某种形式。

明确指明缺陷严重等级和优先等级时刻明确严重等级和优先等级之间的差别。高严重问题可能不值得解决，小装饰性问题可能被当作高优先级

描述 (Description) ，简洁、准确，完整，揭示缺陷实质，记录缺陷或缺陷出现的位置描述要准确反映缺陷的本质内容，简短明了。为了便于在软件缺陷管理数据库中寻找制定的测试缺陷，包含缺陷发生时的用户界面(UI)是个良好的习惯。例如记录对话框的标题、菜单、按钮等控件的名称

短行之间使用自动数字序号，使用相同的字体、字号、行间距短行之间使用自动数字序号，使用相同的字体、字号、行间距，可以保证各条记录格式一致，做到规范专业。

每一个步骤尽量只记录一个操作保证简洁、条理井然，容易重复操作步骤。

确认步骤完整，准确，简短保证快速准确的重复缺陷，“完整”即没有缺漏，“准确”即步骤正确，“简短”即没有多余的步骤。

根据缺陷，可选择是否进行图象捕捉为了直观的观察缺陷或缺陷现象，通常需要附加缺陷或缺陷出现的界面，以图片的形式作为附件附着在记录的“附件”部分。为了节省空间，又能真实反映缺陷或缺陷本质，可以捕捉缺陷或缺陷产生时的全屏幕，活动窗口和局部区域。为了迅速定位、修正缺陷或缺陷位置，通常要求附加中文对照图。 附加必要的特殊文档和个人建议和注解如果打开某个特殊的文档而产生的缺陷或缺陷，则必须附加该文档，从而可以迅速再现缺陷或缺陷。有时，为了使缺陷或缺陷修正者进一步明确缺陷或缺陷的表现，可以附加个人的修改建议或注解。

检查拼写和语法缺陷在提交每条缺陷或缺陷之前，检查拼写和语法，确保内容正确，正确的描述缺陷。

尽量使用短语和短句，避免复杂句型句式软件缺陷管理数据库的目的是便于定位缺陷，因此，要求客观的描述操作步骤，不需要修饰性的词汇和复杂的句型，增强可读性。以上概括了报告测试缺陷的规范要求，随着软件的测试要求不同，测试者经过长期测试，积累了相应的测试经验，将会逐渐养成良好的专业习惯，不断补充新的规范书写要求。此外，经常阅读、学习其他测试工程师的测试缺陷报告，结合自己以前的测试缺陷报告进行对比和思考，可以不断提高技巧。

缺陷描述内容缺陷描述的内容可以包含缺陷操作步骤，实际结果和期望结果。操作步骤可以方便开发人员再现缺陷进行修正，有些开发的再现缺陷能力很差，虽然他明白你所指的缺陷，但就是无法再现特别是对系统不熟悉的新加入开发人员，介绍步骤可以方便他们再现。实际结果可以让开发明白错误是什么，期望结果可以让开发了解正确的结果应该是如何。

21:黑盒测试和白盒测试是软件测试的两种基本方法，请分别说明各自的优点和缺点!

答：黑盒测试的优点有：比较简单，不需要了解程序内部的代码及实现;与软件的内部实现无关; 从用户角度出发，能很容易的知道用户会用到哪些功能，会遇到哪些问题;基于软件开发文档，所以也能知道软件实现了文档中的哪些功能;在做软件自动化测试时较为方便。

黑盒测试的缺点有：不可能覆盖所有的代码，覆盖率较低，大概只能达到总代码量的30%;自动化测试的复用性较低。

白盒测试的优点有：帮助软件测试人员增大代码的覆盖率，提高代码的质量，发现代码中隐 藏的问题。

白盒测试的缺点有：程序运行会有很多不同的路径，不可能测试所有的运行路径;测试基于代码，只能测试开发人员做的对不对，而不能知道设计的正确与否，可能会漏掉一些功能需求;系统庞大时，测试开销会非常大。

22：如何测试一个纸杯?

答：功能度：用水杯装水看漏不漏;水能不能被喝到

安全性：杯子有没有毒或细菌

可靠性：杯子从不同高度落下的损坏程度

可移植性：杯子在不同的地方、温度等环境下是否都可以正常使用

兼容性：杯子是否能够容纳果汁、白水、酒精、汽油等

易用性：杯子是否烫手、是否有防滑措施、是否方便饮用

用户文档：使用手册是否对杯子的用法、限制、使用条件等有详细描述

疲劳测试：将杯子盛上水(案例一)放24小时检查泄漏时间和情况;盛上汽油(案例二)放24小时检查泄漏时间和情况等

压力测试：用根针并在针上面不断加重量，看压强多大时会穿透

23：测试计划工作的目的是什么?测试计划文档的内容应该包括什么?其中哪些是最重要的?

答:　软件测试计划是指导测试过程的纲领性文件：

领导能够根据测试计划进行宏观调控，进行相应资源配置等

测试人员能够了解整个项目测试情况以及项目测试不同阶段的所要进行的工作等

便于其他人员了解测试人员的工作内容，进行有关配合工作

包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

测试计划编写6要素(5W1H)：

why——为什么要进行这些测试;

what—测试哪些方面，不同阶段的工作内容;

when—测试不同阶段的起止时间;

where—相应文档，缺陷的存放位置，测试环境等;

who—项目有关人员组成，安排哪些测试人员进行测试;

how—如何去做，使用哪些测试工具以及测试方法进行测试

测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。所以其中最重要的是测试测试策略和测试方法(最好是能先评审)。

24：黑盒测试的测试用例常见设计方法都有哪些?请分别以具体的例子来说明这些方法在测试用例设计工作中的应用。

答：　1)等价类划分： 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.

2)边界值分析法：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.因此针对各种边界情况设计测试用例,可以查出更多的错误.

使用边界值分析方法设计测试用例,首先应确定边界情况.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.

3)错误猜测法：基于经验和直觉推测程序中所有可能存在的各种错误, 从而有针对性的设计测试用例的方法.

错误推测方法的基本思想: 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,根据他们选择测试用例. 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.

4)因果图方法：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,但未考虑输入条件之间的联系, 相互组合等. 考虑输入条件之间的相互组合,可能会产生一些新的情况. 但要检查输入条件的组合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. 因此必须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. 这就需要利用因果图(逻辑模型). 因果图方法最终生成的就是判定表. 它适合于检查程序输入条件的各种组合情况.

5)正交表分析法：可能因为大量的参数的组合而引起测试用例数量上的激增，同时，这些测试用例并没有明显的优先级上的差距，而测试人员又无法完成这么多数量的测试，就可以通过正交表来进行缩减一些用例，从而达到尽量少的用例覆盖尽量大的范围的可能性。

6)场景分析方法：指根据用户场景来模拟用户的操作步骤，这个比较类似因果图，但是可能执行的深度和可行性更好。

7)状态图法：通过输入条件和系统需求说明得到被测系统的所有状态，通过输入条件和状态得出输出条件;通过输入条件、输出条件和状态得出被测系统的测试用例。

8)大纲法：大纲法是一种着眼于需求的方法，为了列出各种测试条件，就将需求转换为大纲的形式。大纲表示为树状结构，在根和每个叶子结点之间存在唯一的路径。大纲中的每条路径定义了一个特定的输入条件集合，用于定义测试用例。树中叶子的数目或大纲中的路径给出了测试所有功能所需测试用例的大致数量。

25：详细的描述一个测试活动完整的过程。(供参考，本答案主要是瀑布模型的做法)

答：项目经理通过和客户的交流，完成需求文档，由开发人员和测试人员共同完成需求文档的评审，评审的内容包括：需求描述不清楚的地方和可能有明显冲突或者无法实现的功能的地方。项目经理通过综合开发人员，测试人员以及客户的意见，完成项目计划。然后SQA进入项目，开始进行统计和跟踪

开发人员根据需求文档完成需求分析文档，测试人员进行评审，评审的主要内容包括是否有遗漏或双方理解不同的地方。测试人员完成测试计划文档，测试计划包括的内容上面有描述。

测试人员根据修改好的需求分析文档开始写测试用例，同时开发人员完成概要设计文档，详细设计文档。此两份文档成为测试人员撰写测试用例的补充材料。

测试用例完成后，测试和开发需要进行评审。

测试人员搭建环境

开发人员提交第一个版本，可能存在未完成功能，需要说明。测试人员进行测试，发现BUG后提交给BugZilla。

开发提交第二个版本，包括Bug Fix以及增加了部分功能，测试人员进行测试。

重复上面的工作，一般是3-4个版本后BUG数量减少，达到出货的要求。

如果有客户反馈的问题，需要测试人员协助重现并重新测试。

26：BUG管理工具的跟踪过程(用禅道为例子)

答：　测试人员发现了BUG，提交到Bugzilla中，状态为new，BUG的接受者为开发接口人员

开发接口将BUG分配给相关的模块的开发人员，状态修改为已分配，开发人员和测试确认BUG，如果是本人的BUG，则设置为接收;如果是别的开发人员的问题，则转发出去，由下一个开发人员来进行此行为;如果认为不是问题，则需要大家讨论并确认后，拒绝这个BUG，然后测试人员关闭此问题。

如果开发人员接受了BUG，并修改好以后，将BUG状态修改为已修复，并告知测试在哪个版本中可以测试。

测试人员在新版本中测试，如果发现问题依然存在，则拒绝验证;如果已经修复，则关闭BUG。

27：您认为在测试人员同开发人员的沟通过程中，如何提高沟通的效率和改善沟通的效果?维持测试人员同开发团队中其他成员良好的人际关系的关键是什么?

答：尽量面对面的沟通，其次是能直接通过电话沟通，如果只能通过Email等非及时沟通工具的话，强调必须对特性的理解深刻以及能表达清楚。

运用一些测试管理工具如TestDirector进行管理也是较有效的方法，同时要注意在TestDirector中对BUG有准确的描述。

在团队中建立测试人员与开发人员良好沟通中注意以下几点：

一真诚、二是团队精神、三是在专业上有共同语言、四是要对事不对人，工作至上

当然也可以通过直接指出一些小问题，而不是进入BUG Tracking System来增加对方的好感。

28：你对测试最大的兴趣在哪里?为什么?

回答这个面试题，没有固定统一的答案，但可能是许多企业都会问到的。提供以下答案供考：

答：　最大的兴趣，感觉这是一个有挑战性的工作;

测试是一个经验行业，工作越久越能感觉到做好测试的难度和乐趣

通过自己的工作，能使软件产品越来越完善，从中体会到乐趣

回答此类问题注意以下几个方面：

尽可能的切合招聘企业的技术路线来表达你的兴趣，例如该企业是数据库应用的企业，那么表示你的兴趣在数据库的测试，并且希望通过测试提升自己的数据库掌握能力。

表明你做测试的目的是为了提升能力，也是为了更好的做好测试;提升能力不是为了以后转开发或其他的，除非用人企业有这样的安排。

不要过多的表达你的兴趣在招聘企业的范畴这外。比如招聘企业是做财务软件的，可是你表现出来的是对游戏软件的兴趣;或招聘是做JAVA开发的，而你的兴趣是在C类语言程序的开发。

29：你自认为测试的优势在哪里?

答：该面试也没有固定不变的答案，但可参考以下几点，并结合自身特点：

有韧性、有耐心、做事有条理性、喜欢面对挑战、有信心做好每一件事情、较强的沟通能力、从以前的经理处都得到了很好的评价表明我做的很好

问：简述你在以前的工作中做过哪些事情，比较熟悉什么。参考答案如下。

答：　我过去的主要工作是系统测试和自动化测试。在系统测试中，主要是对BOSS系统的业务逻辑功能，以及软交换系统的Class 5特性进行测试。性能测试中，主要是进行的压力测试，在各个不同数量请求的情况下，获取系统响应时间以及系统资源消耗情况。自动化测试主要是通过自己写脚本以及一些第三方工具的结合来测试软交换的特性测试。

在测试中，我感觉对用户需求的完全准确的理解非常重要。另外，就是对BUG的管理，要以需求为依据，并不是所有BUG均需要修改。

测试工作需要耐心和细致，因为在新版本中，虽然多数原来发现的BUG得到了修复，但原来正确的功能也可能变得不正确。因此要注重迭代测试和回归测试。

30：在C/C++中static有什么用途?(请至少说明两种)

答：　1)在函数体，一个被声明为静态的变量在这一函数被调用过程中维持其值不变。

在模块内(但在函数体外)，一个被声明为静态的变量可以被模块内所用函数访问，但不能被模块外其它函数访问。它是一个本地的全局变量。

在模块内，一个被声明为静态的函数只可被这一模块内的其它函数调用。那就是，这个函数被限制在声明它的模块的本地范围内使用

31：引用与指针有什么区别?

答：　1) 引用必须被初始化，指针不必。

引用初始化以后不能被改变，指针可以改变所指的对象。

不存在指向空值的引用，但是存在指向空值的指针。

33：Internet采用哪种网络协议?该协议的主要层次结构?Internet物理地址和IP地址转换采用什么协议?

答：TCP/IP协议主要层次结构为： 应用层/传输层/网络层/数链路层。

ARP (Address Resolution Protocol)(地据址解析协议)

34:说说你对集成测试中自顶向下集成和自底向上集成两个策略的理解，要谈出它们各自的优缺点和主要适应于哪种类型测试;

答：1.自顶向下集成

优点：较早地验证了主要控制和判断点;按深度优先可以首先实现和验证一个完整的软件功能;功能较早证实，带来信心;只需一个驱动，减少驱动器开发的费用;支持故障隔离。

缺点：柱的开发量大;底层验证被推迟;底层组件测试不充分。

适应于产品控制结构比较清晰和稳定;高层接口变化较小;底层接口未定义或经常可能被修改;产口控制组件具有较大的技术风险，需要尽早被验证;希望尽早能看到产品的系统功能行为。

2、自底向上集成

优点：对底层组件行为较早验证;工作最初可以并行集成，比自顶向下效率高;减少了桩的工作量;支持故障隔离。

缺点：驱动的开发工作量大;对高层的验证被推迟，设计上的错误不能被及时发现。

适应于底层接口比较稳定;高层接口变化比较频繁;底层组件较早被完成。

35、软件验收测试包括正式验收测试、alpha测试、beta测试三种测试。

36、系统测试的策略有很多种的，有性能测试、负载测试、强度测试、易用性测试、安全测试、配置测试、安装测试、文档测试、故障恢复测试、用户界面测试、恢复测试、分布测试、可用性测试。

37、设计系统测试计划需要参考的项目文档有软件测试计划、软件需求工件、和迭代计划

38：通过画因果图来写测试用例的步骤为\_\_\_、\_\_\_、\_\_\_、\_\_\_及把因果图转换为状态图共五个步骤。 利用因果图生成测试用例的基本步骤是：

答：§ 分析软件规格说明描述中，哪些是原因(即输入条件或输入条件的等价类)，哪些是结果(即输出条件)，并给每个原因和结果赋予一个标识符。

§ 分析软件规格说明描述中的语义，找出原因与结果之间，原因与原因之间对应的是什么关系? 根据这些关系，画出因果图。

§ 由于语法或环境限制，有些原因与原因之间，原因与结果之间的组合情况不可能出现。为表明这些特殊情况，在因果图上用一些记号标明约束或限制条件。 § 把因果图转换成判定表。

§ 把判定表的每一列拿出来作为依据，设计测试用例。

39：请说出这些测试最好由那些人员完成，测试的是什么?

答：代码、函数级测试一般由白盒测试人员完成，他们针对每段代码或函数进行正确性检验，检查其是否正确的实现了规定的功能。

模块、组件级测试主要依据是程序结构设计测试模块间的集成和调用关系，一般由测试人员完成。

系统测试在于模块测试与单元测试的基础上进行测试。了解系统功能与性能，根据测试用例进行全面的测试。

40：设计测试用例时应该考虑哪些方面，即不同的测试用例针对那些方面进行测试?

答：设计测试用例时需要注意的是，除了对整体流程及功能注意外，还要注意强度测试、性能测试、压力测试、边界值测试、稳定性测试、安全性测试等多方面。(测试用例需要考虑的四个基本要素是输入、输出、操作和测试环境;另外，测试用例需要考虑的是测试类型(功能、性能、安全……)，这部分可以参照TP做答。此外，还需要考虑用例的重要性和优先级)

41： 在windows下保存一个文本文件时会弹出保存对话框，如果为文件名建立测试用例，等价类应该怎样划分?

答：单字节，如A;双字节， AA、我我;特殊字符 /‘。‘;、=-等;保留字，如com;文件格式为8.3格式的;文件名格式为非8.3格式的;/,\*等九个特殊字符。

42：假设有一个文本框要求输入10个字符的邮政编码，对于该文本框应该怎样划分等价类?

答：特殊字符，如10个\*或￥;英文字母，如ABCDefghik;小于十个字符，如123;大于十个字符，如11111111111;数字和其他混合，如123AAAAAAA;空字符;保留字符

43：软件测试项目从什么时候开始，?为什么?

答:软件测试应该在需求分析阶段就介入,因为测试的对象不仅仅是程序编码,应该对软件开发过程中产生的所有产品都测试,并且软件缺陷存在放大趋势.缺陷发现的越晚,修复它所花费的成本就越大.

44：什么是回归测试?

答：回归测试: (regression testing): 回归测试有两类：用例回归和错误回归;用例回归是过一段时间以后再回头对以前使用过的用例在重新进行测试，看看会重新发现问题。错误回归，就是在新版本中，对以前版本中出现并修复的缺陷进行再次验证，并以缺陷为核心，对相关修改的部分进行测试的方法。

45：单元测试、集成测试、系统测试的侧重点是什么?

答：单元测试针对的是软件设计的最小单元–程序模块(面向过程中是函数、过程;面向对象中是类。),进行正确性检验的测试工作,在于发现每个程序模块内部可能存在的差错.一般有两个步骤:人工静态检查\动态执行跟踪

集成测试针对的是通过了单元测试的各个模块所集成起来的组件进行检验,其主要内容是各个单元模块之间的接口,以及各个模块集成后所实现的功能.

系统测试针对的是集成好的软件系统，作为整个计算机系统的一个元素,与计算机硬件\外设\某些支持软件\数据和人员等其他系统元素结合在一起,要在实际的运行环境中,对计算机系统进行一系列的集成测试和确认测试.

46：一个测试工程师应具备那些素质?

答：1、责任心2、沟通能力3、团队合作精神4、耐心、细心、信心5、时时保持怀疑态度，并且有缺陷预防的意识6、具备一定的编程经验

47：你所了解的的软件测试类型都有哪些，简单介绍一下。

答：按测试策略分类：1、静态与动态测试2、黑盒与白盒测试 3、手工和自动测试 4、冒烟测试 5、回归测试;

按测试阶段分类：单元测试、集成测试、系统测试;

其他常见测试方法：1、功能测试 2、性能测试 3、压力测试 4、负载测试 5、易用性测试 6、安装测试 7、界面测试 8、配置测试 9、文档测试 10、兼容性测试 11、安全性测试 12、恢复测试

48：你认为做好测试计划工作的关键是什么?

答：　明确测试的目标，增强测试计划的实用性

编写软件测试计划得重要目的就是使测试过程能够发现更多的软件缺陷，因此软件测试计划的价值取决于它对帮助管理测试项目，并且找出软件潜在的缺陷。因此，软件测试计划中的测试范围必须高度覆盖功能需求，测试方法必须切实可行，测试工具并且具有较高的实用性，便于使用，生成的测试结果直观、准确

坚持“5W”规则，明确内容与过程

“5W”规则指的是“What(做什么)”、“Why(为什么做)”、“When(何时做)”、“Where(在哪里)”、“How(如何做)”。利用“5W”规则创建软件测试计划，可以帮助测试团队理解测试的目的(Why)，明确测试的范围和内容(What)，确定测试的开始和结束日期(When)，指出测试的方法和工具(How)，给出测试文档和软件的存放位置(Where)。

采用评审和更新机制，保证测试计划满足实际需求

测试计划写作完成后，如果没有经过评审，直接发送给测试团队，测试计划内容的可能不准确或遗漏测试内容，或者软件需求变更引起测试范围的增减，而测试计划的内容没有及时更新，误导测试执行人员。

分别创建测试计划与测试详细规格、测试用例

应把详细的测试技术指标包含到独立创建的测试详细规格文档，把用于指导测试小组执行测试过程的测试用例放到独立创建的测试用例文档或测试用例管理数据库中。测试计划和测试详细规格、测试用例之间是战略和战术的关系，测试计划主要从宏观上规划测试活动的范围、方法和资源配置，而测试详细规格、测试用例是完成测试任务的具体战术。

49：您认为做好测试用例设计工作的关键是什么?

答：白盒测试用例设计的关键是以较少的用例覆盖尽可能多的内部程序逻辑结果

黑盒法用例设计的关键同样也是以较少的用例覆盖模块输出和输入接口。不可能做到完全测试，以最少的用例在合理的时间内发现最多的问题

50：你的测试职业发展目标是什么?

答：测试经验越多，测试能力越高。所以我的职业发展是需要时间累积的，一步步向着高级测试工程师奔去。而且我也有初步的职业规划，前3年累积测试经验，不断的更新自己改正自己，做好测试任务。

51：测试结束的标准是什么?

答：从微观上来说，在测试计划中定义，比如系统在一定性能下平稳运行72小时，目前Bug Tracking System中，本版本中没有一般严重的BUG，普通BUG的数量在3以下，BUG修复率90%以上等等参数，然后由开发经理，测试经理，项目经理共同签字认同版本Release。

如果说宏观的，则是当这个软件彻底的消失以后，测试就结束了。

52：一套完整的测试应该由哪些阶段组成?

答：可行性分析、需求分析、概要设计、详细设计、编码、单元测试、集成测试、系统测试、验收测试

53：您是否了解以往所工作的企业的软件开发过程?如果了解，请试述一个完整的开发过程需要完成哪些工作?分别由哪些不同的角色来完成这些工作?您在以往的测试工作中都曾经具体从事过哪些工作?其中最擅长哪部分工作?

答：开发过程—需求调研(需求人员)、需求分析(需求人员)、概要设计(设计人员)、详细设计(设计人员)、编码(开发人员)

测试过程—需求评审、系统测试设计、概要设计评审、集成测试设计、详细设计评审、单元测试设计、测试执行

测试工作的整个过程都做过，擅长做测试设计

过程决定质量，软件的过程改进正是为了提高软件的质量，将过往的种种经验教训积累起来。

54：测试用例设计的原则是什么?目前主要的测试用例设计方法有哪些?

答：代表性：能够代表并覆盖各种合理的和不合理、合法的和非法的、边界的和越界的、以及极限的输入数据、操作和环境设置等.

可判定性：即测试执行结果的正确性是可判定的，每一个测试用例都应有相应的期望结果.

可再现性：即对同样的测试用例，系统的执行结果应当是相同的。

方法有等价类、边界值、因果图、状态图、正交法、大纲法

55：面向对象的测试用例设计有几种方法?如何实现?

答：给类中的每个构造函数设计一组测试用例

组合类中的类变量、实例变量

组合类中的各种方法

根据前置条件和后置条件设计测试用例

根据代码设计测试用例

56：LoadRunner分为哪三个模块?请简述各模块的主要功能。

答：Virtual User Generator：用于录制脚步

Mercury LoadRunner Controller：用于创建、运行和监控场景

Mercury LoadRunner Analysis：用于分析测试结果

57：你对测试最大的兴趣在哪里?为什么?

答：　最大的兴趣就是测试有难度，有挑战性!做测试越久越能感觉到做好测试有多难。曾经在无忧测试网上看到一篇文章，是关于如何做好一名测试工程师。一共罗列了11，12点，有部分是和人的性格有关，有部分需要后天的努力。但除了性格有关的1，2点我没有把握，其他点我都很有信心做好它。

刚开始进入测试行业时，对测试的认识是从无忧测试网上了解到的一些资料，当时是冲着做测试需要很多技能才能做的好，虽然入门容易，但做好很难，比开发更难，虽然当时我很想做开发(学校专业课我基本上不缺席，因为我喜欢我的专业)，但看到测试比开发更难更有挑战性，想做好测试的意志就更坚定了。

我觉得做测试整个过程中有2点让我觉得很有难度(对我来说，有难度的东西我就非常感兴趣)，第一是测试用例的设计，因为测试的精华就在测试用例的设计上了，要在版本出来之前，把用例写好，用什么测试方法写?(也就是测试计划或测试策略)，如果你刚测试一个新任务时，你得花一定的时间去消化业务需求和技术基础，业务需求很好理解(多和产品经理和开发人员沟通就能达到目的)，而技术基础可就没那么简单了，这需要你自觉的学习能力，比如说网站吧，最基本的技术知识你要知道网站内部是怎么运作的的，后台是怎么响应用户请求的?测试环境如何搭建?这些都需要最早的学好。至少在开始测试之前能做好基本的准备，可能会遇到什么难题?需求细节是不是没有确定好?这些问题都能在设计用例的时候发现。

第二是发现BUG的时候了，这应该是测试人员最基本的任务了，一般按测试用例开始测试就能发现大部分的bug，还有一部分bug需要测试的过程中更了解所测版本的情况获得更多信息，补充测试用例，测试出bug。还有如何发现bug?这就需要在测试用例有效的情况下，通过细心和耐心去发现bug了，每个用例都有可能发现bug，每个地方都有可能出错，所以测试过程中思维要清晰(测试过程数据流及结果都得看仔细了，bug都在里面发现的)。如何描述bug也很有讲究，bug在什么情况下会产生，如果条件变化一点点，就不会有这个bug，以哪些最少的操作步骤就能重现这个bug，这个bug产生的规律是什么?如果你够厉害的话，可以帮开发人员初步定位问题。

58：您所熟悉的软件测试类型都有哪些?请试着分别比较这些不同的测试类型的区别与联系(如功能测试、性能测试……)

答：测试类型有：功能测试，性能测试，界面测试。

功能测试在测试工作中占的比例最大，功能测试也叫黑盒测试。是把测试对象看作一个黑盒子。利用黑盒测试法进行动态测试时，需要测试软件产品的功能，不需测试软件产品的内部结构和处理过程。采用黑盒技术设计测试用例的方法有：等价类划分、边界值分析、错误推测、因果图和综合策略。

性能测试是通过自动化的测试工具模拟多种正常、峰值以及异常负载条件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，两者可以结合进行。通过负载测试，确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统各项性能指标的变化情况。压力测试是通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大服务级别的测试。

界面测试，界面是软件与用户交互的最直接的层，界面的好坏决定用户对软件的第一印象。而且设计良好的界面能够引导用户自己完成相应的操作，起到向导的作用。同时界面如同人的面孔，具有吸引用户的直接优势。设计合理的界面能给用户带来轻松愉悦的感受和成功的感觉，相反由于界面设计的失败，让用户有挫败感，再实用强大的功能都可能在用户的畏惧与放弃中付诸东流。

区别在于，功能测试关注产品的所有功能上，要考虑到每个细节功能，每个可能存在的功能问题。性能测试主要关注于产品整体的多用户并发下的稳定性和健壮性。界面测试更关注于用户体验上，用户使用该产品的时候是否易用，是否易懂，是否规范(快捷键之类的)，是否美观(能否吸引用户的注意力)，是否安全(尽量在前台避免用户无意输入无效的数据，当然考虑到体验性，不能太粗鲁的弹出警告)?做某个性能测试的时候，首先它可能是个功能点，首先要保证它的功能是没问题的，然后再考虑该功能点的性能测试

59：请试着比较一下黑盒测试、白盒测试、单元测试、集成测试、系统测试、验收测试的区别与联系。

答：　黑盒测试：已知产品的功能设计规格，可以进行测试证明每个实现了的功能是否符合要求。

白盒测试：已知产品的内部工作过程，可以通过测试证明每种内部操作是否符合设计规格要求，所有内部成分是否以经过检查。

软件的黑盒测试意味着测试要在软件的接口处进行。这种方法是把测试对象看做一个黑盒子，测试人员完全不考虑程序内部的逻辑结构和内部特性，只依据程序的需求规格说明书，检查程序的功能是否符合它的功能说明。因此黑盒测试又叫功能测试或数据驱动测试。黑盒测试主要是为了发现以下几类错误：

1、是否有不正确或遗漏的功能?2、在接口上，输入是否能正确的接受?能否输出正确的结果?3、是否有数据结构错误或外部信息(例如数据文件)访问错误?4、性能上是否能够满足要求?5、是否有初始化或终止性错误?

软件的白盒测试是对软件的过程性细节做细致的检查。这种方法是把测试对象看做一个打开的盒子，它允许测试人员利用程序内部的逻辑结构及有关信息，设计或选择测试用例，对程序所有逻辑路径进行测试。通过在不同点检查程序状态，确定实际状态是否与预期的状态一致。因此白盒测试又称为结构测试或逻辑驱动测试。白盒测试主要是想对程序模块进行如下检查：

1、对程序模块的所有独立的执行路径至少测试一遍。

2、对所有的逻辑判定，取“真”与取“假”的两种情况都能至少测一遍。

3、在循环的边界和运行的界限内执行循环体。

4、测试内部数据结构的有效性，等等。

单元测试(模块测试)是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。通常而言，一个单元测试是用于判断某个特定条件(或者场景)下某个特定函数的行为。

单元测试是由程序员自己来完成，最终受益的也是程序员自己。可以这么说，程序员有责任编写功能代码，同时也就有责任为自己的代码编写单元测试。执行单元测试，就是为了证明这段代码的行为和我们期望的一致。

集成测试(也叫组装测试，联合测试)是单元测试的逻辑扩展。它的最简单的形式是：两个已经测试过的单元组合成一个组件，并且测试它们之间的接口。从这一层意义上讲，组件是指多个单元的集成聚合。在现实方案中，许多单元组合成组件，而这些组件又聚合成程序的更大部分。方法是测试片段的组合，并最终扩展进程，将您的模块与其他组的模块一起测试。最后，将构成进程的所有模块一起测试。

系统测试是将经过测试的子系统装配成一个完整系统来测试。它是检验系统是否确实能提供系统方案说明书中指定功能的有效方法。(常见的联调测试)

系统测试的目的是对最终软件系统进行全面的测试，确保最终软件系统满足产品需求并且遵循系统设计。

验收测试是部署软件之前的最后一个测试操作。验收测试的目的是确保软件准备就绪，并且可以让最终用户将其用于执行软件的既定功能和任务。

验收测试是向未来的用户表明系统能够像预定要求那样工作。经集成测试后，已经按照设计把所有的模块组装成一个完整的软件系统，接口错误也已经基本排除了，接着就应该进一步验证软件的有效性，这就是验收测试的任务，即软件的功能性能如同用户所合理期待的那样。

60：当开发人员说不是BUG时，你如何应付?

答：　开发人员说不是bug，有2种情况，一是需求没有确定，所以我可以这么做，这个时候可以找来产品经理进行确认，需不需要改动，3方商量确定好后再看要不要改。二是这种情况不可能发生，所以不需要修改，这个时候，我可以先尽可能的说出是BUG的依据是什么?如果被用户发现或出了问题，会有什么不良结果?程序员可能会给你很多理由，你可以对他的解释进行反驳。如果还是不行，那我可以给这个问题提出来,跟开发经理和测试经理进行确认,如果要修改就改,如果不要修改就不改。其实有些真的不是bug，我也只是建议的方式写进TD中，如果开发人员不修改也没有大问题。如果确定是bug的话，一定要坚持自己的立场，让问题得到最后的确认。

61：什么要在一个团队中开展软件测试工作?

答：因为没有经过测试的软件很难在发布之前知道该软件的质量，就好比ISO质量认证一样，测试同样也需要质量的保证，这个时候就需要在团队中开展软件测试的工作。在测试的过程发现软件中存在的问题，及时让开发人员得知并修改问题，在即将发布时，从测试报告中得出软件的质量情况。

62：一份测试计划应该包括哪些内容?

答：背景、项目简介、目的、测试范围、测试策略、人员分工、资源要求、进度计划、参考文档、常用术语、提交文档、风险分析。

63：针对于软件的行业背景，你如何理解软件的业务?

答：阅读用户手册了解软件的功能和操作流程;看一些业务的专业书籍补充业务知识;如果有用户实际的数据，可以拿实际的数据进行参考;参考以前的用例和BUG报告;在使用软件的过程中多思考;多与产品经理交流。

64：如何定位测试用例的作用?

答：组织性：编写、组织性、功能覆盖、重复性、跟踪、测试确认

65：什么是兼容性测试?请举例说明如何利用兼容性测试列表进行测试。

答：主要验证软件产品在不同版本之间的兼容性。包括向下兼容和交错兼容，向下兼容是测试软件新版本保留它早期版本功能的情况，交错兼容是验证共同存在的两个相关但不相同的产品之间的兼容性。

66：对某软件进行测试，发现在WIN98上运行得很慢，怎么判别是该软件存在问题还是其软硬件运行环境存在问题?

答：看软件的运行环境要求。如果符合要求则是程序存在问题，若不符合要求则是硬件系统存在问题

67：需求测试的注意事项有哪些?

答：是否使用了公司的模板、文档内容是否符合规范、所有的需求是分级是否清析适当、所有的需求是否具有一致性、需求是否可行(即，该需求组合有解决方案)、需求可否用己知的约束来实现、需求是否足够(即，可以把它送到一个规范的开发组织，并有一个生产出所需要产品的合理的可能性)、所有的其它需求是交叉引用是否正确、用户描述是否清楚、是否用客户的语言来描述需求、每个需求描述是否清楚没有岐义，可以移交给一个独立的组去实现时也能理解、是否所有的需求都是可验证的、是否每条需求都具有独立性，即使发生了变化也不会影响其它需求、性能指标是否明确、非功能性需求是否得到充分表现、是否完整列出适用的标准或协议、标准和协议之间是否存在冲突

68：主键、外键的作用，索引的优点与不足?

答：主键：是表中的唯一标示键。作用：保证实体的完整性;加快数据库的操作速度;增加新的表记录时，数据库会自动检索新记录的主键值，不允许该值与其他表中记录的主键重复;数据库会按主键值的顺序显示记录，如果没有设定主键，则按输入的顺序显示记录。

外键：是主键的从属，表示了两个表之间的联系。作用：使用外键可以避免冗余。

索引的优点： 1、通过创建唯一性的索引，可以保证表中数据的唯一性; 2、加速数据的检索速度; 3、加快表与表之间的连接; 4、在使用分组与排序数据检索时，可以显著检索分组与排序的时间; 5、在查询的过程中使用优化隐藏器，提供系统性能。

缺点： 1、创建索引需要时间，且随着数据量的增加而增加; 2、索引需要占用物理空间;

3、当对表中数据进行修改时，索引也要动态维护，降低了数据的维护速度。

69：性能测试的流程?

答：1.测试需求分析2.测试计划制定与评审3.测试用例设计与开发4.测试执行与监控5.分析测试结果6.编写性能测试报告7.测试经验总结

70：简述bug的生命周期?

答：1， 有效地记录BUG 2， 使用BUG模板 3， 评价BUG优先级和严重性 4， BUG的生命 5， 维护BUG数据库

71：缺陷记录应包含的内容?

答：缺陷标识、缺陷类型、缺陷严重程度、缺陷产生可能性、缺陷优先级、缺陷状态、缺陷起源、缺陷来源、缺陷原因;

72： 您所熟悉的软件测试类型都有哪些?请试着分别比较这些不同的测试类型的区别与联系(如功能测试、性能测试……)

答：易用性测试-界面的友好性，操作方便性等。

功能测试-系统中功能性需求的满足

安全性测试-系统是否存在安全隐患和漏洞

性能测试-系统在大并发下的响应速度和健壮性

73：您认为做好测试计划工作的关键是什么?

答：了解项目或系统的业务需求和项目经理协调好，了解项目的进度计划安排情况

74：您认为做好测试用例设计工作的关键是什么?

答：对业务和软件需求非常清楚，可以根据需求不同选择不同的测试用例设计

75：您以往的工作中是否曾开展过测试用例的评审工作?如果有，请描述测试用例评审的过程和评审的内容。

答：评审计划->预审->评审;

评审内容主要是测试用例对软件需求的覆盖程度，对于相关边界是否考虑，是否针对复杂流程准备多套测试数据，是否有专门针对非功能性需求的测试。

76：您认为性能测试工作的目的是什么?做好性能测试工作的关键是什么?

答：关键是测试脚本的录制，测试时候测试环境的干净。

77：.您以往所从事的软件测试工作中，是否使用了一些工具来进行软件缺陷(Bug)的管理?如果有，请结合该工具描述软件缺陷(Bug)跟踪管理的流程。

CQ，也可以使用BugFree等免费工具。

78：您如何看待软件过程改进?在您曾经工作过的企业中，是否有一些需要改进的东西呢?您期望的理想的测试人员的工作环境是怎样的?

答：将先进的经验或思想固化到过程中，通过过程改进和能力提高来改进软件质量。

TCP/IP五层协议：应用层、传输层、网络层、数据链路层、硬件层

好了，今天关于软件测试问题的面试和答案就先分享到这里了

最后祝大家面试顺利！offer成打来！

“哎，小dao，你是不是还漏了点东西没发出来呀？”

“什么东西？”

“你这都是些测试必备技能面试题，还有日常面试题呢，最经典的“你还有什么问题要问吗”也没看到啊”

“奥，那个啊，其实我们早就发出去了，去【软件测试小dao】公众号就可以看到了~”

更多的简历制作和模板以及面试交流技巧教程可以加软件测试技术交流群：1033482984免费领取

————————————————

版权声明：本文为CSDN博主「软件测试小dao」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/m0_57162664/article/details/116657284>

25.你还做过app项目是吧？这个项目你主要负责什么模块？(我说的是注册登录，--||)

26.APP测试主要针对哪些方面进行测试？(主要是系统，安卓或者ios，弱网，强网，安装/卸载。。。)

27.就这两种了类型吗？安卓系统还分很多种啊，比如华为/魅族/三星，这些你们是怎么测的？

28.你针对这款app负责的部分都是怎么编写测试用例的？从哪些方面编写的？

29.这个电商系统当时主要用的哪种数据库类型啊？(项目上写的是mysql,还好我说对了)

30.你主要负责哪些模块呀？(我负责前台页面)

31.你就负责这么点啊？你们组就两个人，那另一个得负责多少啊？(前台页面，以及上面标注的模块，老早之前的一个项目了，模块名称记不太清了)

32.你们也是用xmind 是吧？主要用它来干什么啊？

33.你就这三个项目啊？

34.你们公司制作自己的项目吗？还是还会承接别人的项目？

35.你熟悉linux是吧？你们用linux都做什么呢？(模拟一些测试环境)

36.模拟什么测试环境呢？模拟出来干什么呢？

37.你对数据库得sql语句会用吗？都有哪些啊？(比较多，经常用的是增删改查。。。)

38.你的数据库主要运用在哪些方面呢？在哪里会用到数据库呢？

39.用数据库怎么进行添加数据呢？

40.Oracle数据库也会用吗？(嗯，sql语句基本互通，关联性数据库。。。)

41.接口怎么对接呢？网络方面你熟悉吗？

42.计算机专业大学也会学习网络方面的吧，你的网络什么通不通你是怎么解决的呢？(我根本没听到她什么意思，就说不清楚，答案是：“ping一下啊，这个太基础了，没人能教你，哈哈哈。。。”

43.你抓包工具都用那些？(F12，Fiddler)

44.别的工具没用过吗？什么mster 什么什么没听过得。。。(没用过，我们主要用Fillder)

45.用哪种抓包工具是公司规定的吗？还是你自己决定的？(怎么答得我也忘了)

46.那你觉得抓包工具F12和Fiddler有什么区别呢？(Fiddler工具更好用，方便简洁，内容明了)

47.你们都用抓包工具干什么呢？解决什么问题呢？

48.app自动化测试有接触过吗？Api接口测试有接触过吗？

终于。。。

“嗯，差不多了，我们面试和笔试是一起的，现在是四点半，还有一个半小时，你接下来没什么安排吧？”

“没有”

“带笔了吗？”

“带了”

“好的，跟我来”

然后，把我带到办公区她的座位旁边，给了我两张A4草稿纸，给了我三张正反面的卷子，她在旁边吃着一个酱香饼之类的饼，后来有个女测试在她旁边汇报工作，什么回归测试啊人手不够啊之类的。。。我坐在她的旁边答题，所

1.接口自动化框架怎么做的，实现了哪些功能，用了多久，框架搭建3周？

2.设计框架遇到了哪些问题，都怎么解决的？

3.你认为你的框架有哪些优点（模块化，数据驱动，发送邮件）缺点，还有什么可以优化的地方？

4.你们都怎么做接口测试，都用啥工具？

5.接口之间的数据依赖？

6.框架的架构（用的语言，用的包）？

7.接口自动化中的断言怎么做，根据啥做的断言（状态码，）

—————————————————————————————————————————–

**一、接口自动化框架怎么做的，实现了哪些功能，用了多久，框架搭建3周？**

1、前期先进行调研：测试数据、功能模块都有哪些、报告是什么形式的

2、介绍模块：一共分成四个模块

common：公共方法，封装了confighttp（封装了get、post方法）、封装了readconfig（读取配置）、封装了configEmail（配置邮件属性、发送邮件）、封装了readExcel（读取存放数据的表格）、封装了writeExcel（复制excel数据，写入数据）

testdata：测试数据，用excel维护，数据方便维护，如果公司选择数据库维护，也可以做

testcase：测试用例，采用ddt测试框架，减少代码编写量，通过一个方法实现了全部测试用例，提高了测试代码的复用率，同时可以提高测试脚本的编写效率

report：测试报告，html

3、框架搭建多长时间：3周，边业务测试边搭建

**二、设计框架遇到了哪些问题，都怎么解决的？**

1、遇到了哪些问题：每次执行用例的时候都会产生很多报告，占用地方，采用循环来清理之前的报告

2、之前写的普通框架，一个用例一个case，用ddt框架实现减少case执行多条用例，缺点：依赖excel强，不能实现定制化

**三、你认为你的框架有哪些优点（模块化，数据驱动，发送邮件）缺点，还有什么可以优化的地方？**

优点：采用ddt框架，数据驱动方式，把数据写在excel表格中，代码做封装，用数据驱动测试，两部分相互独立，代码简单好维护，调试方便。

缺点优化：用例量大，效率低，建议后期优化，采用多线程，报告需要手动清理，优化后可以定时清理，增加日志log

**四、你们都怎么做接口测试，都用啥工具？**

用postman,jmeter 等用具进行接口测试，根据接口文档设计测试用例，测试用例包含对数据的校验，对redis（缓存型数据库）的校验，用fiddler做一些断点，抓包，后期回归用自动化测试框架

**五、接口之间的数据依赖？**

数据依赖：cookies、token，把上一个接口返回的值保存成一个变量，代入到下一个接口中

**六、框架的架构（用的语言，用的包）？**

语言：python  包：requests、unittest、os、xrld、xlutils、configparser、ddt

**七、接口自动化中的断言怎么做，根据啥做的断言（状态码）**

根据状态码、实际接口返回的字段值，和数据库或redis进行比对

自动化测试面试题及答案大全（5）

测试喵 2019-04-20 09:17:45 48243 收藏 425

分类专栏： 测试面试经验 文章标签： 测试  自动化测试 测试面试

版权

1.Selenium是否支持桌面应用软件的自动化测试。

Selenium不支持桌面软件的自动化测试，Selenium是根据网页元素的属性才定位元素，而其他桌面软件自动化测试工具是根据桌面元素的位置来定位元素，当然现在也有根据桌面元素的属性来定位的。

2.Selenium是否支持用例的执行的引擎。

引擎好比就是一个发动机。Selenium是没有关于测试用例和测试套件管理和执行的模块。我们需要借助第三方单元测试框架来实现用例管理和用例的执行。例如Java中有Junit或者testNG,Python中有unittest单元测试框架。

3.Seleinum是否有读取excel文件的库

没有，这里需要用到第三方工具。例如Apache POI插件。

4.Selenium有哪些组件？

最早的有Selenium IDE,IDE只支持安装在fiefox上一个插件，支持录制自动化脚本。还有

remote RC,和Grid 和webdriver。我们一般最重要的就是使用webdriver。

5.Selenium有什么限制或者缺陷

除了基于web的软件和mobile的程序，selenium不支持桌面软件自动化测试。软件测试报告，和用例管理只能依赖第三方插件，例如Junit/TestNG和unittest。由于它是免费的软件，所以没有供应商去提供支持和服务，有问题，只能求助selenium社区。还有一个就是，selenium入门门槛可能有点高，需要具备一定编程语言基础的才能玩转。

6.在selenium中，有哪些不同定位元素方法

ID/className/Name/LinkText/PartialLinkText/Xpath/CSS selector

7.什么是imlicitlyWait（隐式等待）

webdriver 会在指定的超时时间范围内不断的查找元素，直到找到元素或超时

8.什么是expliciteWait（显式等待）

通常是我们自定义的一段代码，这段代码用来等待某种条件发生后，再继续执行后续的代码

9.什么是线程等待（硬式等待）

如Thread.sleep(2000),webdriver线程强制休眠2秒钟，2秒过后，再执行后续的代码。

10.什么是pollingEvery

这个是设置个一段时间就去做一件事，例如下面设置隔一秒就去查找元素一次。

WebDriverWait wait = new WebDriverWait(driver,30);

wait.pollingEvery(1, TimeUnit.SECONDS);

driver.findElement(By.xpath(“xxxx”));

11.你写的测试脚本能在不同浏览器上运行吗，支持跨浏览器平台吗

是的，我写的测试用例能在IE，火狐和谷歌这三种浏览器上运行。主要是在windows平台上运行脚本，所以mac的safari浏览器暂时没有写过。主要实现这个跨浏览器的思想就是，把浏览器类型写到配置文件，代码里写if语句去判断配置文件的浏览器的类型，来决定用什么浏览器去执行测试用例。

12.什么是POM，为什么要使用它

POM是Page Object Model的简称，它是一种设计思想，而不是框架。大概的意思是，把一个一个页面，当做一个对象，页面的元素和元素之间操作方法就是页面对象的属性和行为，所以自然而然就用了类的思想来组织我们的页面。一般一个页面写一个类文件，这个类文件包含该页面的元素定位和业务操作方法

为了我们测试用例写的简单，清晰，我们很多时候在页面对象会封装很多业务操作方法，测试脚本只需要调用相关方法就可以。

还有一个可能和这个问题相关的面试题，如果页面元素经常发生需求变化，你是如何做，答案就是采用POM思想。好处就是只要该一个页面，我就去修改这个页面对象的元素定位和相关方法，脚本不需要修改。

13.在你做自动化过程中，遇到了什么问题吗？举例下

这个问题，不管是自动化还是任何工作，都会被问到。主要想知道你是如何解决问题的，从而推断你问题分析和解决的能力。

当然有遇到问题和挑战，主要有以下几点：

频繁地变更UI，经常要修改页面对象里面代码

运行用例报错和处理，例如元素不可见，元素找不到这样异常

测试脚本复用，尽可能多代码复用

一些新框架产生的页面元素定位问题，例如ck编辑器，动态表格等

14.举例一下你遇到过那些异常，在selenium自动化测试过程中

通过这个问题，大概知道你写过多少脚本。写脚本过程最常见的异常就是，这个元素无法找到。

常见的selenium有以下这些：

ElementNotSelectableException ：元素不能选择异常

ElementNotVisibleException ：元素不可见异常

NoSuchAttributeException ：没有这样属性异常

NoSuchElementException：没有该元素异常

NoSuchFrameException ：没有该frame异常

TimeoutException ： 超时异常

Element not visible at this point ：在当前点元素不可见

15 如何处理alert弹窗

我们常见的alert弹窗有两种：基于windows弹窗和基于web页面弹窗

我们知道，webdriver是能够处理alert弹窗的，Selenium提供了Alert这个接口。相关操作代码如下：

// 切换到Alert

Alert alert = driver.switchTo().alert();

// 点击弹窗上确定按钮

alert.accept();

// 点击弹窗的取消按钮

alert.dismiss()

// 获取弹窗上线上的文本文字内容

alert.getText();

// 有些弹窗还支持文本输入，这个可以把要输入字符通过sendkeys方法输入

alert.sendkeys();

16 在selenium中如何处理多窗口？

这个多窗口之间跳转处理，在实际selenium自动化测试经常遇到。就是，你点击一个链接，这个链接会在一个新的tab打开，然后你接下来要查找元素在新tab打开的页面，所以这里需要用到swithTo方法。

需要获取当前浏览器多窗口句柄，然后根据判断跳转新句柄还是旧句柄

17 你查找元素遇到过在Frame里面吗?你是如何处理Frame里面元素定位的？

有时候我们知道元素定位表达式没有问题，但是还是提示no such element，那么我们就需要考虑这个元素是否在frame中。如果在，我们就需要从topwindow，通过swithcTo.Frame()方法来切换到目标frame中，可以通过frame的name和id和索引三种方法来定位frame。

18 怎么验证勾选框是enable/disabled/ checked/Unchecked/ displayed/ not displayed？

通过以下方法来验证元素是enable 还是disable

boolean enabled = driver.findElement(By.xpath(“元素定位表达式”)).isEnabled();

通过以下方法来验证元素是select/check

boolean checked = driver.findElement(By.xpath(“元素定位表达式”)).isSelected();

通过以下方法来验证元素是dispalyed还是not display

boolean displayed = driver.findElement(By.xpath(“元素定位表达式”)).isDisplayed();

19 如何处理下拉菜单？

通常我们也可以通过Click方法来点击下拉菜单里面的元素，还有一种方法，在Selenium中有一个类叫Select，支持这种下拉菜单交互的操作。

基本使用语法是这样的：

Select Se=new Select(element);

Se.selectByIndex(index);

Se.selectByvalue(value);

Se.selectByVisibleText(text);

20 在日历这种web 表单你是如何处理的?

首先要分析当前网页试用日历插件的前端代码，看看能不能通过元素定位，点击日期实现，如果不能，可能需要借助javascript。还有些日历控件一个文本输入框，可以直接sendKeys()方法来实现传入一个时间的数据。

21关闭浏览器中quit和close的区别

简单来说，两个都可以实现退出浏览器session功能，close是关闭你当前聚焦的tab页面，而quit是关闭全部浏览器tab页面，并退出浏览器session。知道这两个区别，我们就知道quit一般用在结束测试之前的操作，close用在执行用例过程中关闭某一个页面的操作。

22 什么是页面加载超时

Selenium中有一个 Page Load wait的方法，有时候，我们执行脚本的速度太快，但是网页程序还有一部分页面没有完全加载出来，就会遇到元素不可见或者元素找不到的异常。为了解决问题，让脚本流畅的运行，我们可以通过设置页面加载超时时间。具体代码是这个：driver.manage().timeouts().pageLoadTimeout(10,TimeUnit.SECONDS);

这行作用就是，如果页面加载超过10秒还没有完成，就抛出页面加载超时的异常。

23.什么是JavaScript Executor，你什么时候会用到这个？

JavaScript Executor是一个接口，给driver对象提供一个执行javaScript并访问和修改前端元素属性和值。

还是有比较多的场景，我们可能或者需要借助javaScript来实现：

1.元素通过现有定位表达式不能够实现点击

2.前端页面试用了ck-editor这个插件

3.处理时间日期插件（可能）

4.生成一个alert弹窗

5.拖拽滚动条

基本语法：

JavascriptExecutor js =(JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

24.在Selenium中如何实现截图，如何实现用例执行失败才截图

在Selenium中提供了一个TakeScreenShot这么一个接口，这个接口提供了一个getScreenshotAs（）方法可以实现全屏截图。然后我们通过java中的FileUtils来实现把这个截图拷贝到保存截图的路径。

代码举例：

File src=((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

try {

// 拷贝到我们实际保存图片的路径

FileUtils.copyFile(src,new File(“C:/selenium/error.png”));

}

catch (IOException e)

{

System.out.println(e.getMessage());

}

如果要实现执行用例发现失败就自动截图，那么我们需要把这个截图方法进行封装。然后在测试代码中的catch代码块去调用这个截图方法。这个我们在POM的框架中一般是把截图方法封装到BasePage这个文件中。

25.在Selenium中如何实现拖拽滚动条？

在Selenium中通过元素定位会自动帮你拖拽到对应位置，所以是没有自带的scoll方法。但是这个是有限制，例如当前页面高度太长，默认是页上半部分，你定位的元素在页尾，这个时候可能就会报元素不可见的异常。我们就需要利用javaScript来实现拖拽页面滚动条。

我们一般可以两个方法去拖拽，一个是根据拖拽的坐标（像素单位），另外一个是根据拖拽到一个参考元素附件。

代码举例（根据元素坐标拖拽）：

JavascriptExecutor jse= (JavascriptExecutor)driver;

jse.executeScript(“window.scrollBy(0,250)”, “”);

26.如何实现文件上传？

我们在web页面实现文件上传过程中，可以直接把文件在磁盘完整路径，通过sendKeys方法实现上传。如果这种方法不能实现上传，我们就可能需要借助第三方工具，我用过一个第三方工具叫autoIT.

还有一个方法是利用robot类

27.如何处理“不受信任的证书”的问题？

例如，在登录12306网站的时候，如果你没有下载和安装过这个网站的根证书，那么你就会遇到打开12306网站提示证书不受信任的拦截页面。

下面举例火狐和谷歌上处理这个问题的基本代码

火狐：

// 创建firefoxprofile

FirefoxProfile profile=new FirefoxProfile();

// 点击继续浏览不安全的网站

profile.setAcceptUntrustedCertificates(true);

// 使用带条件的profile去创建一个driver对象

WebDriver driver=new FirefoxDriver(profile);

Chrome：

// 创建类DesiredCapabilities的对象

DesiredCapabilities cap=DesiredCapabilities.chrome();

// 设置ACCEPT\_SSL\_CERTS 变量值为true

cap.setCapability(CapabilityType.ACCEPT\_SSL\_CERTS, true);

// 新建一个带capability的chromedriver对象

WebDriver driver=new ChromeDriver(cap);

28.什么是Firefox Profile?

Profile是一组文件，主要用来记录用户在火狐浏览器上的私人信息，例如书签，密码，用户首选项，下载文件夹保存路径等。简单来说，你打开火狐浏览器输入about://config，这个页面有些设置选项是可以通过profile来实现修改的。

29.如何实现鼠标悬停，键盘事件和拖拽动作？

在Webdriver中，处理键盘事件和鼠标事件，一般使用Actions类提供的方法，包括鼠标悬停，拖拽和组合键输入。

这里介绍几个方法

方法：clickAndHold()

使用场景：找到一个元素，点击鼠标左键，不放手。自己可以点击鼠标不松开试试这个场景。

方法：contentClick()

使用场景：模拟鼠标右键点击，一般右键会带出菜单来。

方法：doubelClick()

使用场景：模拟鼠标双击

方法：dragAndDrop(source,target)

使用场景：模拟从source这个位置，拖拽一个元素到target位置

键盘事件方法:keyDown(keys.ALT), keyUp(keys.SHIFT)

使用场景：点击键盘事件，分为两个动作，一个点击键盘，第二个动作是释放点击（松开）

30.在执行脚本过程，如何实现当前元素高亮显示？

这个其实就是利用javaScript去修改当前元素的背景颜色来到达高亮显示的效果，

31.如何获取页面标题，悬浮文本和错误文本，并验证？

标题，我们可以通过driver.getTitle()方法来得到一个字符串，然后使用字符串的containts方法或者equals方法去进行断言。

悬浮文本（tooltip），一般是利用Actions类，然后鼠标悬停方法，然后通过getText()方法来得到这个tooltip字符串。

错误信息，直接把这个错误字段先进行定位，然后通过getText()方法拿到错误文本，主要的断言有包含，相等，不相等，不包含，以什么开头等。

32.在selenium自动化测试中，你一般完成什么类型的测试？

主要是冒烟测试和回归测试。回归测试主要写一些功能稳定，容易实现的场景，通过自动化手段去实现，节约测试时间。

33.你是如何管理你的测试用例并执行？

写用例和管理并执行用例，我们都需要借助单元测试框架来实现，如果是Java语言一般有junit和TestNG，如果是python，常见的有unittest。

就你实际情况，说一下。例如我使用TestNG比较多，需要配置testng.xml文件来实现测试用例的执行。有时候需要配置多个testng.xml去实现不同的任务场景。再展开，可能问你一下testng框架的知识点。例如，方法依赖，用例执行优先级，数据源驱动等。

34关于自动化测试报告生成？

我个人一般用TestNG原生的测试报告，也有第三方叫reportNG的插件，不过我没有实际使用过。

————————————————

版权声明：本文为CSDN博主「测试喵」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/qq_33299873/article/details/89412776>

selenium 处理页面下拉选择框

1、 标准下拉框（Select 标签 option 选项）（如下图）

针对标准下拉框，selenium 提供了处理方法，代码如下：

Select sel =new Select(driver.findElement(By.id(“areaID”))); //定义 select 方法

sel.selectByIndex(1); //根据选项的索引定位

sel.selectByValue(“tianjin”); //根据选项的 value 值定位

sel.selectByVisibleText(“陕西省”); //根据选项的 text 文本值定位

2、 非标准下拉框

处理方法：模拟手工测试时的思路，先点击一次使下拉列表展开，在点击下拉选项

代码如下

第一步：

driver.findElement(By.cssSelector("#divselect>cite")).click(); //定位方法仅供参考

第二步：

driver.findElement(By.linkText(“下拉 select 效果”)).click(); //注意，点击下拉选项前建议增

加 1 秒 sleep

————————————————

版权声明：本文为CSDN博主「软件测试开发-虚竹」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/shuaigezhou10086/article/details/109607195>

1 什么是POM，为什么要使用它？

POM是Page Object Model的简称，它是一种设计思想，而不是框架。大概的意思是，把一个一个页面，当做一个对象，页面的元素和元素之间操作方法就是页面对象的属性和行为，所以自然而然就用了类的思想来组织我们的页面。一般一个页面写一个类文件，这个类文件包含该页面的元素定位和业务操作方法

为了我们测试用例写的简单，清晰，我们很多时候在页面对象会封装很多业务操作方法，测试脚本只需要调用相关方法就可以。

2如果页面元素经常发生需求变化，你是如何做?

采用POM思想。好处就是只要改一个页面，我就去修改这个页面对象的元素定位和相关方法，脚本不需要修改。

3 在你做自动化过程中，遇到了什么问题吗？举例下

频繁地变更UI，经常要修改页面对象里面代码

运行用例报错和处理，例如元素不可见，元素找不到这样异常

测试脚本复用，尽可能多代码复用

一些新框架产生的页面元素定位问题，例如ck编辑器，动态表格等

4 举例一下你遇到过那些异常，在selenium自动化测试过程中

ElementNotSelectableException：元素不能选择异常

ElementNotVisibleException ：元素不可见异常

NoSuchAttributeException ：没有这样属性异常

NoSuchElementException：没有该元素异常

NoSuchFrameException ：没有该frame异常

TimeoutException ： 超时异常

Element not visible at this point ：在当前点元素不可见

5 如何处理alert弹窗

我们常见的alert弹窗有两种：基于windows弹窗和基于web页面弹窗

webdriver是能够处理alert弹窗的，Selenium提供了Alert这个接口。

相关操作代码如下：

Alert alert = driver.switchTo().alert() // 切换到Alert

alert.accept() // 点击弹窗上确定按钮

alert.dismiss() // 点击弹窗的取消按钮

alert.getText() // 获取弹窗上线上的文本文字内容

alert.sendkeys() // 有些弹窗还支持文本输入，这个可以把要输入字符通过sendkeys方法输入

6 在selenium中如何处理多窗口？

这个多窗口之间跳转处理，在实际selenium自动化测试经常遇到。就是，你点击一个链接，这个链接会在一个新的tab打开，然后你接下来要查找元素在新tab打开的页面，所以这里需要用到swithTo方法。

需要获取当前浏览器多窗口句柄，然后根据判断跳转新句柄还是旧句柄

7 你查找元素遇到过在Frame里面吗?你是如何处理Frame里面元素定位的？

有时候我们知道元素定位表达式没有问题，但是还是提示no such element，那么我们就需要考虑这个元素是否在frame中。如果在，我们就需要从topwindow，通过swithcTo.Frame()方法来切换到目标frame中，可以通过frame的name、id和index三种方法来定位frame。

8 如何处理下拉菜单？

通常我们也可以通过Click方法来点击下拉菜单里面的元素，还有一种方法，在Selenium中有一个类叫Select，支持这种下拉菜单交互的操作。

基本使用语法是这样的：

Se=new Select(element)

Se.selectByIndex(index)

Se.selectByvalue(value)

Se.selectByVisibleText(text)

1

2

3

4

9 关闭浏览器中quit和close的区别?

close是关闭你当前聚焦的tab页面;

而quit是关闭全部浏览器tab页面，并退出浏览器session；

quit一般用在结束测试之前的操作，close用在执行用例过程中关闭某一个页面的操作

10 什么是imlicitlyWait（隐式等待）

webdriver 会在指定的超时时间范围内不断的查找元素，直到找到元素或超时

11 什么是expliciteWait（显式等待）

通常是我们自定义的一段代码，这段代码用来等待某种条件发生后，再继续执行后续的代码

12 如何实现文件上传？

我们在web页面实现文件上传过程中，可以直接把文件在磁盘完整路径，通过sendKeys方法实现上传

13 如何实现鼠标悬停，键盘事件和拖拽动作？

在Webdriver中，处理键盘事件和鼠标事件，一般使用Actions类提供的方法，包括鼠标悬停，拖拽和组合键输入。

这里介绍几个方法

方法： clickAndHold()

使用场景：找到一个元素，点击鼠标左键，不放手。自己可以点击鼠标不松开试试这个场景。

方法：contentClick()

使用场景：模拟鼠标右键点击，一般右键会带出菜单来。

方法：doubelClick()

使用场景：模拟鼠标双击

方法：dragAndDrop(source,target)

使用场景：模拟从source这个位置，拖拽一个元素到target位置

键盘事件方法:keyDown(keys.ALT), keyUp(keys.SHIFT)

使用场景：点击键盘事件，分为两个动作，一个点击键盘，第二个动作是释放点击（松开）

14 在selenium自动化测试中，你一般完成什么类型的测试？

主要是冒烟测试和回归测试。回归测试主要写一些功能稳定，容易实现的场景，通过自动化手段去实现，节约测试时间。

15 你是如何管理你的测试用例并执行？

通过单元测试框架实现，常见的有unittest、pytest

16 自动化测试报告生成

Allure

17 你会封装自动化测试框架吗？

自动化框架主要的核心框架就是分层+PO模式：

分别为：基础封装层BasePage，PO页面对象层，TestCase测试用例层。然后再加上日志处理模块，ini配置文件读取模块，unittest+ddt数据驱动模块，jenkins持续集成模式组成。

18 自动化测试有误报过bug吗？产生误报怎么办？

有误报过，有时候自动化测试报告中显示发现了bug,实际去通过手工测试去确认又不存在该bug。

误报原因一般是：

1.元素定位不稳定，需要尽量提高脚本的稳定性；

2.开发更新了页面但是测试没有及时更新维护!

19 自动化测试过程中，你遇到了哪些问题，是如何解决的？

1.频繁地变更页面，经常要修改页面对象类里面的代码

2.自动化测试偶尔出现过误报

3.自动化测试结果出现覆盖的情况：Jenkins根据时间建立文件夹

4.自动化测试代码维护比较麻烦

5.自动化测试进行数据库对比数据

20 在上一家公司做自动化测试用的什么框架？

可以说出以下自己擅长的一种：

1.python+selenium+unittest+htmltestrunner

2.python+selenium+pytest+allure

3.robotframework+Selenium2Library

21 遇到frame框架页面怎么处理？

先用driver.switch\_to.frame()跳转进去frame，

然后再操作页面元素，

操作完后使用driver.swith\_to.default\_content()跳转出来

22 遇到alert弹出窗如何处理？

使用driver.switch\_to.alert()方法先跳转到alert弹出窗口

然后再通过accept点击确定按钮；

通过dismiss点击取消难；

通过text获得弹出窗口的文本;

23 什么是断言

断言assert 用于代码中验证实际结果符合预期结果，如果测试用例失败，则抛出异常并且提供断言日志

24 如何提高selenium脚本的执行速度

1、优化等待时间，使用WebDriverWait智能等待代替强制等待sleep和隐式等待imlicitlyWait

2、减少不必要的操作

3、在服务器允许的情况下，使用多线程实现并发执行测试用例

25 怎么对含有验证码的功能进行自动化测试

图像识，难度大，效果不好不推荐

屏蔽验证码，邀请开发处理，在测试环境，预发和正式环境恢复

26 自动化测试分为哪几类

UI自动化：又分为web和和app自动化

接口自动化

27 自动化测试的使用场景？

需求稳定，不会频繁变更。

研发和测试周期长，需要频繁执行回归测试。

需要在多种平台上重复运行相同测试的场景。

某些测试项目，通过手工测试无法实现，或者手工成本太高。

被测软件的开发较为规范，能够保证系统的可测试行

28 请描述一下自动化测试流程？

1.编写自动化测试计划

2.设计自动化测试用例

3.编写自动化测试框架和脚本

4.调试并维护脚本

5.无人值守测试

6.后期脚本维护（添加用例、开发更新版本）

29 一个接口的响应在下一个接口中怎么用？(一个请求依赖另一个请求的返回结果)

cookie 全局变量 反射

存储到excel表，需要时再取

框架里边的期望结果： 查库 依赖用户成功之后

30 web和app自动化有什么不同？

1 启动差别

app端：在执行用例的时候，一部安卓手机同一时刻只能打开一个apk包进行操作

web端：在web端，通过Python多线程(或多进程)同时开启几个浏览器，让selenium对多个浏览器进行操作；

2 安装检查

app端：需要先检查软件是否安装才能进行测试

web端：不需要安装，在浏览器中输入url就可以测试。

3 页面元素操作

app端：需要保证不可见的元素显示在手机页面才能对它进行操作。

web端：如果遇到需要下拉才能加载的页面，可以用js操作滚动条。

4 元素定位

app端：部分定位方式不支持，比如css\_selector和link\_text

web端：name，id，class\_name，css，xpath、link\_text、partrail\_link\_text、tag\_name

5 启动方式

app端：需要制定desired\_caps内容，因为里面包含了设备信息等。

web端：通过启动webdriver不同的浏览器类，获取driver，如webdriver.Chrome()，也可以模拟手机端加载wap页面做wap页面的测试。

31 unitest和pytest框架讲解以及使用的是哪个一个为什么不用另一个？

较unittest，pytest有以下优点

自动发现测试模块、测试方法

断言使用asert+表达式

可以设置会话级、模块级、类级、函数级的fixtures、数据准备+清理工作

有丰富的插件库，目前在300个以上。

32 分别说出web和app元素定位方法

Web：id、xpath、name、class\_name、tag\_name、link\_text、partial\_link\_text、css\_selector

app： id、classname、xpath

33 get和post不同点

GET - 从指定的资源请求数据。请求的数据会附加在URL之后，以?分割URL和传输数据，多个参数用&连接

POST - 向指定的资源提交要被处理的数据。POST请求会把请求的数据放置在HTTP请求包的包体中

34 http和https不同点

1、HTTPS 协议需要到 CA （Certificate Authority，证书颁发机构）申请证书，一般免费证书较少，因而需要一定费用。(以前的网易官网是http，而网易邮箱是 https 。)

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。(无状态的意思是其数据包的发送、传输和接收都是相互独立的。无连接的意思是指通信双方都不长久的维持对方的任何信息。)

35 selenium原理

当使用 Selenium 2.0 启动浏览器时，后台会同时启动基于 WebDriver Wire 协议的 Web Service 作为 Selenium 的 Remote Server，并与浏览器绑定。之后Remote Server 就开始监听 Client 端的操作请求；

执行测试时，测试用例会作为 Client 端，将需要执行的页面操作请求以 Http Request 的方式发送给 Remote Server 。该 Http Request 的 body，是以 WebDriver Wire 协议规定的 JSON 格式来描述需要浏览器执行的具体操作；

Remote Server 接收到请求后，会对请求进行解析，并将解析结果发给 WebDriver，由WebDriver 实际执行浏览器的操作；

WebDriver 可以看做是直接操作浏览器的原生组件（Native Component），所以搭建测试环境时，通常都需要先下载浏览器对应的 WebDriver。

36 appium原理

开源、跨平台的UI自动化测试工具，支持多种语言编写的测试脚本

原理：

test scripts(测试脚本发送一个请求到appium server)

appium server接收到请求后进行解析并把请求转发给 bootstrap.jar。

jar接收到appium的命令，调用UIAutomator命令实现操作

最终结果由bootstrap.jar返回给Appium server。

37自动化测试用到的模块

requests+unittest+ddt+httptestrunner+pymysql+openpyxl+logging 接口自动化

selenium+pytest+allure web自动化

appium+selenium+pytest+allure+yaml app自动化

38 OSI七层模型

物理层、数据链路层、网络层、传输层、会话层、表示层、应用层

39 cookie、session、token各自区别

cookie:在客户端存储在客户端用于存储会话信息的

session：在服务器端，记录用户的请求状态，一般默认时间30min

session\_id会存在cookie中，每次请求cookie中所有信息都会传递给服务器，服务器通过 session\_id来识别是否是同一个用户请求，不是同一个用户的话，就会要求重新登录

token：访问权限

鉴权：访问的接口是否正常，是否非法访问绕过前端。防止跳过页面直接访问接口。token

授权：是否具有访问接口的权限。 唯一全局动态的 。key

40 常用状态码

100系列：请求已收到继续处理；

200系列:表示成功

200：正常，服务器正确响应了请求

300系列：资源重定向；

301:永久重定向；请求的网页已永久移动到新位置

302：2临时重定向；被请求文档已经临时移至别处，此文档新的url在location响应头中给出

303：浏览器对于POST的响应进行重定向至新的url

307：浏览器对于GET的响应重定向至新的url

400系列：客户端错误：

400：错误请求；服务器不理解请求的语法。

401：未授权；如请求参数、方法、格式等

403：拒绝访问；服务器理解客户的请求，但拒绝处理它（没有权限）

404：请求资源不存在

500系列：服务器端出错

500：服务器内部错误

501：尚未实施；服务器不具备完成请求的功能

502:服务器网关错误

503：服务器由于维护或者负载过重未能应答

504请求超时

41 手写adb命令

adb 帮助：adb --help

启动adb 服务：adb start-server

关闭adb 服务：adb kill-server

获取设备号：adb devices

获取系统版本：adb shell getprop ro.build.version.release

发送文件到手机：adb push 电脑端⽂件路径/需要发送的文件 手机端存储的路径

adb push C:\Users\win\Desktop\xx.png /sdcard

从手机拉取文件: adb pull 手机端的路径/拉取文件名 电脑端存储文件路径

adb pull /sdcard/xx.png C:\Users\win\Desktop

查看手机运行日志: adb logcat

进入到手机终端: adb shell

安装app到手机: adb install 路径/xxx.apk

卸载手机app : adb uninstall app

获取app启动包名和启动名(⚠手机需要先打开对应app)

Mac/Linux: adb shell dumpsys window windows | grep mFocusedApp

在 Windows 终端运⾏： adb shell dumpsys window windows | findstr mCurrent

获取app启动时间: adb shell am start -W 包名/.启动名

查看设备ip地址：

adb shell ifconfig wlan0

adb shell netcfg

查看设备cpu信息： adb shell cat /proc/cpuinfo

查看设备内存信息： adb shell cat /proc/meminfo

42 http请求头和响应头

http请求及其结构：

请求信息包含：请求行(request) 请求头部header 、空行和请求数据组成

响应及其结构

响应组成：状态行、响应头报头、空行和响应正文

43 鼠标操作常用函数

context\_click() 右击 --> 此方法模拟鼠标右键点击效果

double\_click() 双击 --> 此方法模拟双标双击效果

drag\_and\_drop() 拖动 --> 此方法模拟双标拖动效果

move\_to\_element() 悬停 --> 此方法模拟鼠标悬停效果

perform() 执行 --> 此方法用来执行以上所有鼠标方法

44 键盘操作常用函数

send\_keys(Keys.BACK\_SPACE) 删除键（BackSpace）

send\_keys(Keys.SPACE) 空格键(Space)

send\_keys(Keys.TAB) 制表键(Tab)

send\_keys(Keys.ESCAPE) 回退键（Esc）

send\_keys(Keys.ENTER) 回车键（Enter）

send\_keys(Keys.CONTROL,'a') 全选（Ctrl+A）

send\_keys(Keys.CONTROL,'c') 复制（Ctrl+C）

send\_keys(Keys.CONTROL,'v') 全选（Ctrl+V）

send\_keys(Keys.CONTROL,'x') 复制（Ctrl+X）

45. 解决手动造数据问题

参数化

手机号：

excel里边存放初始手机号 每次执行完，回写新的手机号(原来号码+1)

每次从数据库里查询最大手机号，在这个基础上加1

变量替换： 数据库依赖关系 ${mobile} ${regtime} ${memberid} ${loanid}

最关键：用例设计、用例参数之间依赖关系

46 你写框架多长时间？

初步模型：1-2周，一个月时间

47 TestCase使用

导入unittest模块、被测文件或其中的类

创建一个测试类，并继承unitest.TestCase

定义测试函数，函数名已test\_开头，测试用例

调用unittest.main()方法运行测试用例

48 Selenium 中如何保证操作元素的成功率？也就是说如何保证我点击的元素一 定是可以点击的？

1.添加元素智能等待时间 driver.implicitly\_wait(30)

2.添加强制等待时间 time.sleep()

3.try 方式进行id,name,clas,x path, css selector不同方式进行定位，如 果第一种失败可以自动尝试第二种

50 你的自动化用例的执行策略是什么？

利用自动化测试工具，经过测试需求分 析；

设计出自动化测试用例；

从而搭建自动化测试的框架，设计与编写自动化脚 本；

验证测试脚本的正确性，最终完成自动化测试测试脚本（即主要功能为测试的应用软件）

输出测试结果

51 常见的 POST 提交数据方式

主要有四种方式：

application/x-www-form-urlencoded；

multipart/form-data；

application/json；

text/xml

52 目前主流的APP自动化测试框架，各个自动化适合的语言

appium macaca、robotium、UiAutomator

53 Selenium有哪几种定位方式？用的最多的是哪种？

8种单元素定位方法，8种对应的多元素定位方法；所有方法都是基于driver.find\_element()和driver.find\_elements()方法，如下：

from selenium import webdriver

from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.find\_element\_by\_id('id') # 通过元素id属性定位元素

driver.find\_element\_by\_name('name') # 通过元素名字属性定位元素

driver.find\_element\_by\_tag\_name('tag\_name') # 通过元素的标签名称定位元素

driver.find\_element\_by\_class\_name('class\_name') # 通过元素的类名称定位元素

driver.find\_element\_by\_link\_text('link\_text') # 通过链接元素的完整显示文字定位元素

driver.find\_element\_by\_partial\_link\_text('partial\_link\_text') # 通过链接元素的部分显示文字定位元素

driver.find\_element\_by\_xpath('xpath') # 通过xpath表达式定位元素

driver.find\_element\_by\_css\_selector('css\_selector') # 通过css表达式定位元素

driver.find\_elements\_by\_id('id') # find\_elements\_by\_xxx系列也有8种，可以定位多个满足条件的元素

driver.find\_element(By.ID, 'id') # 每一个find\_element\_by\_xxx方法都是基于find\_element方法的

driver.find\_elements(By.ID, 'id') # find\_element()方法的复数形式，可以定位多个满足条件的元素

1

2

3

4

5

6

7

8

9

10

11

12

13

14

为了保证代码的统一性，使用的最多的是driver.find\_element\_by\_xpath()这个方法，传入定位器，即使我们使用id作为定位器，仍然写成基于id的xpath定位器表达式，如下：

baidu\_yi\_xia\_locator = ‘//input[@id="su"]’ # 百度一下按钮的定位器，写成了xpath，但基于的是id这个属性

baidu\_yi\_xia = driver.find\_element\_by\_xpath(baidu\_yi\_xia\_locator) #定位元素，并返回给变量

baidu\_yi\_xia.click() # 点击百度一下

1

2

3

54 UI自动化能发现多少Bug

UI自动化的目的不是为了发现多少Bug，主要是为了减轻重复的基础操作和线上监控的作用

55 monkey属于自动化吗？

monkey不属于严格意义上的自动化，monkey是生成用户或系统的伪随机事件，在屏幕上触发随机点击事件

56 你们一般对什么case会进行自动化，自动化一般在哪个阶段进行

主要是主流程中比较容易实现的进行自动化，一般在集成阶段进行该版本的自动化监控，平常的话会一直跑线上监控的

57 app自动化你们一般用什么工具定位元素？

Uiautomatorview和appium的客户端

58 您需要一台服务器机器来运行Appium上的测试吗？

不需要服务器机器在Appium上运行测试。 Appium促进了一个2层架构，其中测试机连接到运行Appium的测试服务器并自动化整个事情。您可以在运行测试的同一台机器上运行Appium。

59 使用Appium可能遇到的错误是什么？

错误1：需要以下所需的功能，但不提供：设备名称，platformName

错误2：找不到adb。请使用Android SDK根目录路径设置ANDROID\_HOME环境变量

错误3：openqa.selenium.SessionNotCreatedException：无法创建新的会话

错误4：如何在移动应用程序中查找DOM元素或XPath？

60 简述Appium的原理？

Appium是使用Node.js平台编写的“HTTP Server”，并使用Webdriver JSON线协议驱动iOS和Android会话。

在初始化Appium Server之前，必须在系统上预先安装Node.js 当Appium被下载并安装时，在我们的机器上设置一个暴露REST API的服务器

它从客户端接收连接和命令请求，并在移动设备（Android / iOS）上执行该命令，

它响应HTTP响应。

再次，为了执行此请求，它使用移动测试自动化框架来驱动应用程序的用户界面。 框架像Apple Instruments for iOS（仅适用于Xcode 3.0或更高版本的OS X v10.5及更高版本）适用于Android API的Google UIAutomator 16级或更高版本Selendroid for Android API等级在15以下。

61 如何提高selenium脚本的执行速度？

如网速、操作步骤的繁琐程度、页面加载的速度、在脚本中设置的等待时间、运行脚本的线程数等。所以不能单方面追求运行速度的，要确保稳定性，能稳定地实现回归测试才是关键。

减少操作步骤，如经过三四步才能打开我们要测试的页面的话，我们就可以直接通过网址来打开，减少不必要的操作。

中断页面加载，如果页面加载的内容过多，我们可以查看一下加载慢的原因，如果加载的内容不影响我们测试，就设置超时时间，中断页面加载。

在设置等待时间的时候，可以sleep固定的时间，也可以检测某个元素出现后中断等待也可以提高速度。

四，配置testNG实现多线程。在编写测试用例的时候，一定要实现松耦合，然后在服务器允许的情况下，尽量设置多线程运行，提高执行速度。

62 什么是持续集成？

持续集成源于极限编程(XP)，是一种软件实践，软件开发过程中集成步骤是一个漫长并且无法预测的过程。集成过程中可能会爆发大量的问题，因此集成过程需要尽可能小而多，实际上持续集成讲的是不断的去做软件的集成工作。持续集成，最简单的形式是包括一个监控版本控制（SVN等等）变化的工具。当变化被发觉时，这个工具可以自动的编译并测试你的应用

63 什么是page object设计模式？

通过分离测试对象和测试脚本的抽象来实现的

64 你觉得自动化测试最大的缺陷是什么？

不稳定

可靠性

不易维护

成本与收益

65 Selenium是否支持桌面应用软件的自动化测试。

Selenium不支持桌面软件的自动化测试，Selenium是根据网页元素的属性才定位元素，而其他桌面软件自动化测试工具是根据桌面元素的位置来定位元素，当然现在也有根据桌面元素的属性来定位的。

66 BDD是什么？你了解多少？TDD是什么？

BDD：行为驱动开发（Behavior Driven Development）

TDD：测试驱动开发（Test-Driven Development）

67 selenium是否可以直接读取Excel表中测试用例，来执行相关测试

可以的，需要借助第三方库

68 Selenium有哪些组件？

最早的有Selenium IDE,IDE只支持安装在fiefox上一个插件，支持录制自动化脚本。还有

remote RC、Grid 、webdriver。我们一般最重要的就是使用webdriver。

69 如果元素定位中遇到iFrame内嵌框架，你是如何定位的？如果没有遇到id属性和name属性为空的情况，又是如何处理的？

第一个问题：遇到iFrame内嵌框架里的任何元素，和以往一样写出定位器就可以了，只是在执行脚本的时候，定位器是正确的，但是仍然脚本执行失败，报错“无法找到元素”。原因就是因为这个元素被嵌在了iFrame内嵌框架中（我们也叫子框架），所以需要在定位元素前，先写以下代码，作用是从当前的主框架切换到内嵌框架中，有多种方式：

# 切入frame有3种方式

# (1) 通过frame的id属性或name属性

driver.switch\_to.frame('layui-layer-iframe1')

# (2) 通过frame的自然序号，从0开始（第1个iframe）

driver.switch\_to.frame(0)

# (3) 通过定位器，配合find\_element()方法，定位到框架元素后，再把这个定位到的框架元素入参iframe\_loc = '//iframe[@id="layui-layer-iframe1"]'

driver.switch\_to.frame(driver.find\_element\_by\_xpath(iframe\_loc))

1

2

3

4

5

6

7

第二个问题：没有id或者name属性，只需要用xpath写出表达式，表达式中利用元素的其他单个或多个属性的组合，只要能够保证元素的唯一性即可，如下例子:

//input[@value="百度一下"]

1

70 明明自己定位的元素是对的，执行自动化测试脚本时却报错，这时你有几种方法解决此问题？请写出你的解决方法。

(1)元素在iFrame里：先切入iFrame

(2)元素在打开的新窗口里：先切入新窗口

(3)元素在新跳转的页面里，但是因为各种原因，新页面跳转很慢，已经超过了Selenium中对于元素定位的最大等待时间：增加隐式等待时间或对这个元素进行智能等待

隐式等待的代码如下：

from selenium import webdriver

driver = webdriver.Chrome()

# 设置全局等待（即：隐式等待），注意只需要设置一次

driver.implicitly\_wait(60) # 延长到60秒

显示等待（智能等待）的代码如下：

from selenium import webdriver

from selenium.webdriver.support.wait import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

driver = webdriver.Chrome()

mobile\_phone\_loc = (By.XPATH, '//input[@name="mobilephone"]')

elem = WebDriverWait(driver, 60).until(EC.element\_to\_be\_clickable(mobile\_phone\_loc))

elem.send\_keys('13812345678')

1

2

3

4

5

6

7

8

9

10

11

12

71 简单说出如何用自动化测试脚本实现遍历复选框点击功能（要求全部勾上）。

xPathRadio = '//input[@type="radio"]'

# radiobox复选框对象我们必须使用find\_elements方法去定位多个元素（复选框就是多元素）

radioboxes = driver.find\_elements\_by\_xpath(xPathRadio)

# 使用循环遍历的方式，逐个点击这些定位到的元素

for radiobox in radioboxes:

radiobox.click()

1

2

3

4

5

6

72 写一个自动化脚本，语言不限，要求每执行一次脚本随机生成一个手机号码。

import random # 导入随机数模块

# 一开始，手机号是空的

mobile\_phone = ''

# 在中国，手机号码的第一位都是1

phone\_num\_1 = '1'

# 把第一位生成的数字拼接到手机号

mobile\_phone = mobile\_phone + phone\_num\_1

# 根据不同的运营商，手机号码的第2位和第3位都是有固定值的，这里随便列举了几个，放入列表

phone\_num\_2\_to\_3 = ['38', '82', '88', '36', '30']

# choices方法可以随机抽取列表里的元素，从而生成第二三位手机号码，并拼接到手机号

phone\_num\_2\_to\_3 = random.choices(phone\_num\_2\_to\_3)[0]

mobile\_phone = mobile\_phone + phone\_num\_2\_to\_3

# 最后8个数字，循环生成

for i in range(8):

# randint方法可以随机产生0~9的值，但是为了拼接字符串，我们用str()方法转换

phone\_num = str(random.randint(0, 9))

# 每一次循环都会生成一个数字，并且继续拼接到现有的电话号码里

mobile\_phone = mobile\_phone + phone\_num

# 打印最终生成的手机号码

print(mobile\_phone)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

73 你对单元测试框架了解多少

unittest、testng、nose、pytest

————————————————

版权声明：本文为CSDN博主「JSon liu」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_45912307/article/details/109438188>

json和python中字典的区别？

Json是轻量级的数据交互格式，以key-value的键值对形式来保存数据，结构清晰，可以说是目前互联网项目开发中最常用的一种数据交互格式。

字典，同样是以key-value的键值对来保存数据，是python中的一种数据类型。

你做接口自动化测试时，测试数据放哪里？

1) 对于一些基础配置比如数据库配置可以放到properties文件(yaml文件)

2) 接口测试需要用到的数据可以放数据库表，也可以放到excel中。

什么是数据驱动？

数据驱动就是通过excel，或者xml，数据库等数据结构来维护测试数据，通过相应的技术去处理，拿到测试数据用于测试，从而实现测试数据与代码的解耦，数据驱动测试的本质是高级的参数化

excel中如何设计你的用例？

1) 所有的接口信息维护在一个表单

2) 关于接口的用例数据维护在一个表单

3) 接口传参一列来传，通过构造json格式的字符串即可解决传多个参数的问题，同时提升了用例的可维护性

excel中的数据怎么拿出来？

通过第三方库openpyxl来完成，项目只需要引入openpyxl库，通过它提供的api即可完成对于excel的操作，甚至于我们自己也可以做一些二次封装，使得操作更方便。一行数据就是一个用例。

如果有些接口做了鉴权(需要鉴定权限)，怎么实现这些接口的自动化测试？

首先需要明确接口是通过什么机制来实现的鉴权

1) 如果是通过会话id(JSESSIONID)来实现鉴权的，那么先调用完登录接口--》拿到它返回的这个数据--》存储起来--》后面再调用其他接口的时候直接带上这个数据过去就可以了。

2) 如果是通过token来实现鉴权的，那么同样只需要在调用完登录接口--》拿到它返回的这个数据--》存储起来，后面操作同上。

先回答上面的一个基本实现步骤，然后再补充以下一些实现细节：

a. 怎么取到这个JSESSIONID或者token

这个信息怎么取首先得明白它是怎么返回来的，可能为以下两种方式：

1) 登录接口的响应头

2) 登录接口的响应报文

需要找开发确认，不同方式不同取法。

b. 怎么存储这个JSESSIONID或者token

1) 文件中(excel、properties文件都可以)

2) 数据库

3) 代码里面自己定义的公共静态变量，作为一个共享的数据

c. 请求其他需要鉴权的接口时，怎么带过去这个数据

先找开发人员确认，再调用需要鉴权的这些接口怎么携带这个鉴权的信息(票)。一般不外乎这么几种：

1) 拼接在url上传过去用于服务器验证

2) 设置到请求体中

3) 通过请求头带过去

弄明白怎么传过去以后，就设法从保存了这个鉴权信息的地方取出来--》传过去

什么是会话？

所谓会话就是session，而session是一种服务器的缓存技术，由服务器端创建，用户登录了系统，一般就会把登录用户的信息保存到这个session对象中，然后把此会话对应的id(JSESSIONID)返回，因此大部分系统都会用到session机制来实现鉴权。Session保存数据方式也是以key-value的形式。

什么是token？

所谓token其实就是服务器端返回的一个字符串(有点类似于：xys73494954sdhcdr83435这么一串)，这个数据是基于什么算法生成的需要找开发确认，一般这个数据是唯一的，服务器每次返回的token都会不一样。Token之所以可以用来做鉴权，原理如下：用户a调用了登录接口--》登录了系统b--》服务器端生成了一个唯一的token信息(假设为c)，然后会拿用户的编号id(假设为d)做了一个映射：c-d，然后将这样的映射关系存到数据库或redis等缓存中，然后再返回这个token给客户端--》客户端再调用其他需要鉴权的接口时，只需要将缓存起来的这个token带过去验证--》服务器根据此token检查是否有登录用户信息来判断此请求是否是一个已登录授权的用户。(这里客户端怎么取到这个token，怎么存，再次请求时怎么带过去，上面接口鉴权部分已解释)。

某个接口请求，参数依赖上个接口的返回数据，怎么处理(接口关联)？

场景：b接口在调用的时候参数依赖于a接口的返回数据。

处理方案：

a接口调用完--》从响应中取出b依赖的这个参数--》保存起来--》b调用的时候，取出这个信息--》当做参数传过去即可。

保存方案：

参考问题5中的数据保存方式

接口调用依赖第三方的接口b，而b接口还未提供给你，如何处理？

通过mock框架搭建自己的mock系统/平台(其实就是自己开发接口)，模拟这个第三方接口mock一个出来即可，这样就解决了这个依赖。

接口测试的垃圾数据如何清理？

可以自己准备单独的环境(独立的数据库)，这样接口调用产生的数据可以在整个套件执行完后通过执行sql脚本来完成数据删除

做接口自动化时怎么做的数据验证？

数据验证主要包括两个方面：

1) 接口响应报文的验证

利用jsonpath取出响应数据(大多数情况是json)中的关键字段的值，通过Assert断言这些关键字段的值是否符合预期。预期值和jsonpath取值表达式可以配置在excel中。

2) 数据库表数据的验证

做表数据验证要先明白两点：

a. 表数据验证首先得明白这个接口的基本业务，会影响到哪些表，哪些字段的数据，然后才能准备查询sql去有针对性的查数据。

b. 并不是所有用例的执行都会操作数据库，影响到表数据，因此只有影响到了表数据的用例，才准备查询脚本。

具体做法：

Excel用例准备aa,bb,cc三列，aa传脚本，bb，cc两列分别用来回写用例执行前的sql查询结果和用例执行后的sql查询结果，两个数据对比即可知道数据变化是否满足预期。

如何提升测试套件的执行效率？

1.减少excel读写，减少不必要的io操作

2.excel数据读取测试数据，全部存储到静态变量中)。

3.excel数据回写，尽量通过批量方式去写。

你用过单元测试框架吗，是什么框架，有哪些常用操作？

用过，unittest 或者 pytest，这些测试框架都支持我们去定义测试套件，达到管理我们测试用例代码的作用，同时这些测试框架提供的一些丰富注解不仅能很方便的控制测试用例的执行顺序来控制整个测试流程，还能为各种测试场景的实现提供支撑。

常用操作：

seutp,teardown，适合单个测试用例的初始化、环境清理。

seutpClass,teardownClass：适合测试类的全局初始化、环境清理。

assertXXXX: 各种断言方式

testLoader的discover: 用来发现测试用例

testSuite的addTests: 用来装载测试用例

@ddt  数据驱动

htmlTestRunner：用来生成html测试报告

说说你对接口的理解？

接口是服务，是功能点实现，是数据传递的通道，也是服务器端的一个实现了某种协议(比如http协议..)和映射机制(当访问一个url时就会通过服务器端的映射处理机制解析后，落到对应的处理函数)的函数，接口参数就是函数的参数，接口的响应数据就是函数的返回值

请简单说一说get、post这两种请求方法的区别？

Get：

a. 一般对于从服务器取数据的请求可以设置为get方式

b. Get方式在传递参数的时候，一般都会把参数直接拼接在url上(例如:http://xxx?id=1&type=2)

c. Get请求方法可以传递的参数数据量是有限的(因为一般参数是拼接在url上，而浏览器对于url的长度是有限制的)

d. Get请求因为是直接将数据拼接在url上，所以安全性不如post强(相对)，但是get的执行效率要快于post

Post：

a. 一般往服务器提交数据的请求会设置为post方式

b. Post方式传递参数一般都会把参数放在请求体，而不会拼接在url

c. Post请求方法可以提交的数据量不受限制

d. Post请求传参相对于get要安全(相对不是绝对)，但是post的执行效率不如get。

在浏览器中输入URL，回车之后发生了什么？

1.解析地址

2. 浏览器查询，是否存在浏览器缓存中，如果没有

3.Tcp连接，想服务器发送请求

4.服务器返回接口，并且在浏览器中添加缓存

5.前端渲染页面

======================================

(1)URL 解析

(2)DNS 查询

(3)TCP 连接

(4)处理请求

(5)接受响应

(6)渲染页面

————————————————

版权声明：本文为CSDN博主「雪灵谷自然地理」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_28778267/article/details/114914110>

**自动化测试Selenium最新面试题和对应答案！**



[zhusongziye](https://blog.csdn.net/zhusongziye) 2018-05-21 20:34:18  8168  收藏 44

分类专栏： [Selenium自动化测试](https://blog.csdn.net/zhusongziye/category_7131148.html) 文章标签： [自动化测试Selenium最新面试题和对应答案！](https://so.csdn.net/so/search/s.do?q=%E8%87%AA%E5%8A%A8%E5%8C%96%E6%B5%8B%E8%AF%95Selenium%E6%9C%80%E6%96%B0%E9%9D%A2%E8%AF%95%E9%A2%98%E5%92%8C%E5%AF%B9%E5%BA%94%E7%AD%94%E6%A1%88%EF%BC%81&t=blog&o=vip&s=&l=&f=&viparticle=)

问题1：Selenium是什么，流行的版本有哪些？

Selenium是基于Web的最流行的UI自动化测试工具。它提供了一组支持多种平台的公开API（例如Linux，Windows，Mac OS X等）。此外，像Google Chrome，Mozilla Firefox，Internet Explorer和Safari等所有现代浏览器都可以用来运行Selenium测试。它也涵盖了Android平台，其中Appium是实现Selenium Webdriver界面的工具，用于移动自动化。

值得注意的是，除了许多后来的小型版本之外，硒还有三个主要版本：

Selenium 1.0或Selenium RC，于2004年初发布，提供了一个使用服务器与浏览器交换命令和响应的API集。

Selenium 2.0或Selenium Webdriver，在2011年中推出，并在Selenium功能中引入了一系列重大改进。这些新的API完全取代了服务器组件，并与目标浏览器本地交互。

Selenium 3.0，这个版本是在2016年末发布的大版本。它带来的主要变化是引入Webdriver API的W3C规范，用于浏览器自动化。也就是说，每个主要的浏览器都会有自己的Webdriver API来实现功能。

问题2：你如何从命令行启动Selenium RC？

// 简单的启动Selenium RC的方法是

java -jar selenium-server.jar

// 在浏览器中运行一套Selenese脚本

java -jar selenium-server.jar -htmlSuite

问题3：在我的机器端口4444不是免费的。我怎样才能使用另一个端口？

//你可以在运行selenium服务器时指定端口为 -

Java -jar selenium-server.jar -port 5555

问题4：什么是Selenium Server，它与Selenium Hub有什么不同？

Selenium Server是使用单个服务器作为测试节点的一个独立的应用程序。 Selenium hub代理一个或多个Selenium的节点实例。一个hub 和多个node被称为Selenium grid。运行SeleniumServer与在同一主机上用一个hub和单个节点创建de Selenium grid类似。

问题5：你如何从Selenium连接到数据库？

Selenium是一个Web UI自动化工具。它不提供任何API来建立数据库连接。这取决于你使用Selenium进行自动化的编程语言。在下面的例子中，我们假设正在使用Java。

一个Connection对象表示与数据库的连接。当我们使用连接方法连接到一个数据库时，我们创建了一个连接对象，它代表了与数据库的连接。单个数据库可能有一个连接或多个连接，还可能有多个连接到不同的数据库上。

我们可以使用Connection对象来做以下事情：

* 创建用于执行SQL语句的Statement，PreparedStatement和CallableStatement对象。
* 可以帮助我们提交或回滚一个JDBC事务。
* 如果你想知道连接到的数据库或数据源信息，Connection对象通过使用DatabaseMetaData就可以收集有关数据库或数据源的信息。
* 可以帮助我们关闭数据源。Connection.isClosed() 方法只有在调用了**Connection.close()**时才返回true 。此方法用于关闭所有连接。

首先我们需要通过使用**DriverManager.getConnection()**方法，建立与数据库的连接。这个方法接受一个包含URL的字符串。DriverManager类尝试查找可以连接到由字符串URL表示的数据库的驱动程序。每当调用**getConnection()**方法时，DriverManager类都会检查可以连接到URL中指定的数据库的所有已注册的Driver类的列表。

句法：

String url = "jdbc: odbc: makeConnection";

Connection con = DriverManager.getConnection(url, "userID", "password");

问题6：Selenium RC中定位器方法有哪些？

1. ID
2. Name
3. CSS (Cascade Stylesheet)
4. XPATH (Relative XPath and Absolute XPath)
5. Dom

问题7：你如何验证多个页面上存在的一个对象？

可以使用下面的Selenium命令来检查：

assertTrue(selenium.isElementPresent(locator));

问题8：XPath中使用单斜杠和双斜杠有什么区别？

* 如果XPath是从文档节点开始，它将允许创建“绝对”路径表达式。

**例如** “/ html / body / p”匹配所有的段落元素。

* 如果XPath在文档中的任意位置开始进行选择匹配，那么它将允许创建“相对”路径表达式。

**例如** “// p”匹配所有的段落元素。

问题9：如何编写Selenium IDE / RC的用户扩展？

用户扩展（UX）存储在Selenium IDE或Selenium RC用来激活扩展的单独文件中。它包含用JavaScript编写的函数定义。

因为Selenium的核心是用JavaScript开发的，所以要符合原语言的标准规则来创建扩展。要创建一个扩展，我们必须用下面的设计格式来编写函数。

// 样例

Selenium.prototype.doFunctionName = function(){

}

函数名称前面的“do”告诉Selenium这个函数可以被调用为一个步骤命令，而不是作为内部函数或私有函数被调用。

问题10：如何在页面加载成功后验证元素的存在？

它可以通过下面的代码行来实现。

只需一点时间（以秒为单位）来检查元素，如下所示：

public void waitForElementPresent(String element, int timeout) throws Exception {

for (int second = 0;; second++) {

if (second >= timeout)

fail("Timeout. Unable to find the Specified element" + element);

try {

if (selenium.isElementPresent(element))

break;

} catch (Exception e) {

}

Thread.sleep(1000);

}

}

问题11：你对Selenium Grid有什么了解？它提供了什么功能？

Selenium Grid是一款利用现有计算基础架构大幅加速Web应用程序功能测试的工具。允许测试者轻松地在多台机器上并行运行多个测试，并且可以在异构环境中运行。

基于优秀的Selenium Web测试工具，Selenium Grid允许测试者并行运行多个Selenium Remote Control实例。更好的是，它集成显示所有Selenium远程控制，所以不必担心实际的基础设施。Selenium Grid将运行Selenium测试套件所需的时间，缩短到Selenium实例的单个实例运行时间的一小点。

问题12：如何从你的Java Class启动Selenium服务器？

try {

seleniumServer = new SeleniumServer();

seleniumServer.start();

} catch (Exception e) {

e.printStackTrace();

}

问题13：Selenium中有哪些验证点？

Selenium主要有三种验证点 -

* 检查页面标题
* 检查某些文字
* 检查某些元素（文本框，下拉菜单，表等）

问题14：什么是XPath？什么时候应该在Selenium中使用XPath？

XPath是一种在HTML / XML文档中定位的方法，可用于识别网页中的元素。如果没有与页面上的元素相关联的名称/ ID，或者名称/ ID的一部分是常量，则必须使用XPath。

绝对路径用 - / 单斜杠

相对路径用 - // 双斜杠

ID，类，名称也可以用于XPath ：

* //input[@name=’q’]
* //input[@id=’lst-ib’]
* //input[@class=’ lst’]

如果id / name / class的一部分是常量 ：

* //input[contains(@id,’lst-ib’)

问题15：什么是Selenium的CSS定位器策略？用例子来解释。

CSS位置策略可以与Selenium一起使用来定位元素，它使用CSS定位方法，其中 -

绝对路径用 - （空格符号）

相对路径用 - >表示

ID，类，名称也可以用于XPath：

* css=input[name=’q’]
* css=input[id=’lst-ib’] or input#lst-ib
* css=input[class=’lst’] or input.lst

如果id / name / class只有一部分是常量：

* css=input[id\*=’lst-ib’)]

使用内部文本的元素位置策略：

* css = a:contains(‘log out’)

问题16：当有很多定位器时，如ID、名称、XPath、CSS定位器，我应该使用哪一个？

如果有唯一的名称或标识符可用，那么应该使用它们来代替XPath和CSS定位器。如果没有，那么CSS定位器应该被优先考虑，因为在大多数现代浏览器中，它们的评估速度比XPath更快。

问题17：在硒中处理多个弹出窗口的机制是什么？

可以使用命令**getWindowHandles()**来处理多个弹出窗口。

然后将所有窗口名称存储到**Set**变量中并将其转换为数组。

接下来，通过使用数组索引，导航到特定的窗口。

driver.switchTo().window(ArrayIndex);

问题18：你如何处理使用Selenium的Ajax控件？

来看一个例子。假如一个文本框是一个Ajax控件，当我们输入一些文本时，它会显示自动建议的值。

处理这样的控件，需要在文本框中输入值之后，捕获字符串中的所有建议值；然后，分割字符串，取值就好了。

**Part2. WebDriver相关面试问题和答案**

问题19：Selenium Webdriver优于Selenium RC的优点是什么？

Selenium RC的架构相当复杂，WebDriver的架构比Selenium RC简单些。

* Selenium RC比较慢，因为它使用了另外一个名为Selenium Core的JavaScript程序。相反，WebDriver比Selenium RC更快，因为它直接与浏览器对话，并使用浏览器自己的引擎来进行控制。
* 像其他JavaScript代码一样，Selenium Core可以访问禁用的元素。Webdriver以更现实的方式与页面元素进行交互。
* Selenium RC的API集已经有所改进，但是仍有经常让人困惑的冗余部分。WebDriver API更简单，不包含任何冗余或混淆的命令。
* Selenium RC无法支持无头HtmlUnit浏览器。它需要一个真正的、可见的浏览器来操作。Web Driver可以支持无头HtmlUnit浏览器。
* Selenium RC内置了测试结果生成器，并自动生成测试结果的HTML文件。Web驱动程序没有自动生成测试结果文件的内置命令。

问题20：“GET”和“NAVIGATE”方法的主要区别是什么？

Get方法能获得一个页面进行加载、或获取页面源代码、或获取文本，就这三。而Navigate将通过刷新，回退，前进的方式导航。

**例如 -**如果我们想要前进，并做一些功能，并返回到主页。

这可以通过调用**< navigate()>**方法来实现。

**driver.get()** 方法会等到整个页面被加载后才可以，而**driver.navigate()**只是重定向到该网页，并不会等待。

问题21：隐式等待与显式等待有什么不同？

**隐式等待**是设置的全局等待，分为1、页面加载超时等待 ；2、页面元素加载超时；3、异步脚本超时。如果是页面元素超时，设置等待时间，是对页面中的所有元素设置加载时间。隐式等待是其实可以理解成在规定的时间范围内，浏览器在不停的刷新页面，直到找到相关元素或者时间结束。

**显式等待**只是用于特定搜索的一个计时器。它的可扩展性更强，你可以设置它来等待任何条件。通常情况下，可以使用一些预先构建的条件来等待元素变得可点击，可见，不可见等，或者只是编写适合需求的条件。

问题22：你将如何处理Selenium WebDriver中的警报/弹出窗口？

有两种类型的警报通常被引用。

* 基于Windows的警报弹出窗口
* 基于Web的警报弹出窗口

基于Web的警报弹出窗口。

1. WebDriver为用户提供了一种使用Alert界面处理这些弹出窗口的非常有效的方法。
2. void dismiss() - 一旦出现弹出窗口，dismiss()方法就会点击“Cancel”按钮。
3. void accept() - 只要弹出窗口出现，accept()方法就会点击“Ok”按钮。
4. String getText() - getText()方法返回警告框中显示的文本。
5. void sendKeys(String stringToSend) - sendKeys()方法将指定的字符串模式输入到警告框中。

基于Windows的警报弹出窗口。

处理基于windows的弹出窗口总是有点棘手，因为我们知道Selenium是一个自动化测试工具，它只支持Web应用程序测试，也就是说，它不支持基于Windows的应用程序，窗口警报就是其中之一。

1. Robot class是基于Java的实用程序，它模拟键盘和鼠标操作，并可以有效地用于处理基于windows的弹出与键盘事件的帮助。
2. KeyPress和KkeyRelease方法可以分别模拟用户按下和释放键盘上某个键的操作。

问题23：如何使用Selenium WebDriver截图？



问题24：如何在WebDriver/如何管理HTTPS中的安全连接错误中解决Firefox中的SSL证书问题？



问题25：如何解决IE中的SSL认证问题？

// 打开浏览器后添加下面的命令

driver.navigate().to(“javascript:document.getElementById(‘overridelink’).click()”);

问题26：Selenium WebDriver中的可用定位器是什么？

1. ID,
2. Name,名称
3. CSS,
4. XPath,
5. Class name,
6. TagName,
7. LinkText, 链接文本
8. Partial Link Text.部分链接文本

问题27：如何处理WebDriver中的AJAX控件？

**AJAX**代表异步JavaScript和XML。它不依赖于创建有效的XML所需的打开和关闭标签的额外开销。大部分时间WebDriver自动处理Ajax控件和调用。如果不能处理的话，可以按照下面的方式来处理。

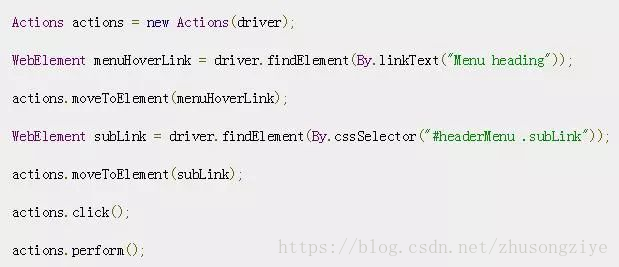
//Waiting for Ajax Control

WebElement AjaxElement = (new WebDriverWait(driver,

10)).until(ExpectedConditions.presenceOfElementLocated(By.("")));

问题28：如何在标题菜单的子菜单项上执行鼠标移动操作？

应该首先移动菜单标题，然后移至弹出菜单项并单击它。不要忘记在最后调用actions.perform() 。以下是一些示例Java代码：



**Part3. 一般框架面试问题和答案**

问题29：大致分类和比较TDD/BDD和DDD框架？

你可能听说过所有的这些缩写词。在这里会简要地解释它们，以及它们在系统测试生命周期中如何发挥作用的。

TDD - 测试驱动开发。

也被称为测试驱动设计，是一个软件开发的方法，在源代码上重复进行单元测试。写测试、看它失败、然后重构。这个概念是，先编写测试，然后来检查我们写的代码是否正常工作。每次测试后，重构完成，然后再次执行相同或类似的测试。该过程需要重复多次，直到每个单元在功能上按预期工作。TDD是由XP引入的。

BDD - 行为驱动开发。

行为驱动的开发将TDD的一般技术和原理与领域驱动设计的思想相结合。其目的是帮助人们设计系统（即开发人员）确定合适的测试来编写测试 - 即反映利益相关者所期望行为的测试。

DDD域驱动的开发。

DDD将业务领域概念映射到软件工件中。DDD框架提供以下好处：

* 帮助团队在业务和IT利益相关者之间建立一个共同的模型
* 该模型是模块化的，可扩展的，易于维护，该设计反映了一种商业模式。
* 它提高了业务领域对象的可重用性和可测试性。

问题30：什么是数据驱动框架？它与关键字驱动框架有什么不同？

**数据驱动框架。**

在这个框架中，测试用例逻辑驻留在测试脚本中。测试数据被分离并保存在测试脚本之外。测试数据是从外部文件（Excel文件）中读取的，并被加载到测试脚本中的变量中。变量用于输入值和验证值。

**关键字驱动。**

关键字/表驱动框架需要开发数据表和关键字。它们独立于执行它们的测试自动化工具。可以使用或不使用应用程序来设计测试。在关键字驱动的测试中，被测试的应用程序的功能记录在一个表格中，以及每个测试的分步说明。

问题31：解释使用TestNG而不是JUnit框架的好处？

**TestNG相较于Junit的优势：**

1. 在JUnit中，我们必须声明@BeforeClass和@AfterClass，这是JUnit中的一个约束，而在TestNG中没有像这样的约束。
2. TestNG提供了更多的setUp / tearDown级别。1.@ Before/AfterSuite 2.@Before/AfterTest 3.@Before/AfterGroup
3. TestNG中不需要扩展任何类。
4. TestNG中没有方法名称约束，就像JUnit一样。
5. 在TestNG中，我们可以告诉测试一个方法依赖于另一个方法，而在JUnit中这是不可能的。
6. 测试用例的分组在TestNG中可用，而JUnit中则不可用。执行可以基于组完成。例如，如果你已经定义了许多案例，并通过将2个组分别定义为“离职“与”回归”隔离。如果你只是想执行“理智”的情况，那就告诉TestNG执行“理智”。TestNG将自动执行属于“离职”组的案例。
7. 另外，TestNG支持并行测试用例执行。

问题32：与@Test注释相关的TestNG参数的目的是什么？

在TestNG中，参数是修改注释功能的关键字。

问题33：可以使用TestNG运行一组测试用例吗？

是的，TestNG框架支持在测试组的帮助下执行多个测试用例。

它提供了以下选项来运行特定组中的测试用例。

如果想基于回归测试或冒烟测试等其中一个组来执行测试用例，那么：

@Test(groups = {“regression-tests”, “smoke-tests”})

问题34：WebDriver哪个实现是最快的，为什么？

WebDriver的最快的实现是 **HTMLUnitDriver**。

原因是**HTMLUnitDriver**不会在浏览器中执行测试。相反，它使用简单的HTTP请求 - 响应机制来运行测试用例。

这种方法比需要启动浏览器来测试执行的方式要快得多。

问题35：是否可以在Selenium 2.0中使用Selenium RC API？

是的，可以用Selenium 2.0来模拟Selenium 1.0 API（即RC）。但并不是所有的Selenium 1.0方法都支持。

为了达到这个目的，需要从WebDriver获取Selenium实例并使用Selenium方法。

在Selenium 2.0中模拟Selenium 1.0时，方法执行速度也可能会变慢。

问题36：可以在Java，Dot Net或Ruby中使用Selenium Grid吗？

* 使用Java，可以利用TestNG的并行测试功能来驱动Selenium Grid测试。
* 使用.Net，可以使用“Gallio”并行执行测试。
* 使用Ruby，可以使用“DeepTest”来分发测试。

python selenium自动化测试面试题\_selenium 自动化测试面试题及答案

weixin\_39860064 2020-12-20 23:10:42 228 收藏 1

文章标签： python selenium自动化测试面试题

版权

工程师小C的小店我也想开通小店

Python编程三剑客：Python编程从入门到实践第2版+快速上手第2版+极客编程（套装共3册）

作者：[美] 埃里克·马瑟斯（Eric Matthes）

出版社：人民邮电出版社

好评：100.0%

销售量：20

￥149

更多

1、selenium中如何判断元素是否存在？

- isElementPresent

2、selenium中hidden或者是display ＝ none的元素是否可以定位到？

- 不能

3、selenium中如何保证操作元素的成功率？也就是说如何保证我点击的元素一定是可以点击的？

- 添加元素智能等待时间 driver.implicitly\_wait(30)

- try 方式进行 id,name,clas,x path, css selector 不同方式进行定位，如果第一种失败可以自动尝试第二种

-Selenium保证元素成功率是通过元素的定位，当然它的定位方法很多，一定能有合适的。但是在自动化工程的实施过程中，高质量的自动化测试不是只有测试人员保证的。需要开发人员规范开发习惯，如给页面元素加上唯一的name,id等，这样就能大大地提高元素定位的准确性。当然如果开发人员开发不规范，我们在定位元素的时候尽量使用相对地址定位，这样能减少元素定位受页面变化的影响。只要我们元素定位准确，就能保证我的每一个操作符合我的预期

4、如何提高selenium脚本的执行速度？

- Selenium脚本的执行速度受多方面因素的影响，如网速，操作步骤的繁琐程度，页面加载的速度，以及我们在脚本中设置的等待时间，运行脚本的线程数等。所以不能单方面追求运行速度的，要确保稳定性，能稳定地实现回归测试才是关键。

我们可以从以下几个方面来提高速度：

一，减少操作步骤，如经过三四步才能打开我们要测试的页面的话，我们就可以直接通过网址来打开，减少不必要的操作。

二，中断页面加载，如果页面加载的内容过多，我们可以查看一下加载慢的原因，如果加载的内容不影响我们测试，就设置超时时间，中断页面加载。

三，在设置等待时间的时候，可以sleep固定的时间，也可以检测某个元素出现后中断等待也可以提高速度。

四，配置testNG实现多线程。在编写测试用例的时候，一定要实现松耦合，然后在服务器允许的情况下，尽量设置多线程运行，提高执行速度。

5、用例在运行过程中经常会出现不稳定的情况，也就是说这次可以通过，下次就没办法通过了，如何去提升用例的稳定性？

- time.sleep( )

- driver.implicitly\_wait(30)

- 多用 try 捕捉，处理异常

-此时我们要分析出不稳定的原因，然后有针对性的去解决问题。主要有以下几个方面 ：

一，网速问题：有的时候网页加载的比较慢，在程序执行的时候要操作的元素没有显示出来。这种情况比较常见，运行一次网速好的时候通过了，再运行一次，页面没有打开，就不通过了。为了提高稳定性，我们只能牺牲运行时间了，在经常检测失败的元素前加上等待时间，等要操作的元素出现之后再执行下面的操作。

二，Selelnium的原因：Selenium1.0和2.0还是有区别的，有些儿函数在2.0下运行确实有时而有效，时面无效。如果mouseover()函数，就是这种情况， 我们需要避免使用这类的函数。

三，多线程的时候，测试用例间相互影响。虽然多线程的时候运行速度比较快，但是如果用例之间的耦合性没有设计好，也会影响的，如果用例A先于用例B执行的时候，就会影响到用例B；反之则没有问题。这种情况，如果你的自动化测试工程打算多线程的时候，提前就要把测试用例测试的耦合度比较松，尽量没有任何关系，因为多线程的执行顺序是不受控制的。

6、你的自动化用例的执行策略是什么？

- 自动化测试用例的执行策略是要看自动化测试的目的，通常有如下几种策略：

一，自动化测试用例是用来监控的，在此目的下，我们就把自动化测试用例设置成定时执行的，如果每五分钟或是一个小时执行一次，在jenkins上创建一个定时任务即可。

二，必须回归的用例。有些儿测试用例，如BVT测试用例，我们在公司产品任何变动上线之前都需要回归执行。那我们就把测试用例设置成触发式执行，在jenkins上将我们的自动化测试任务绑定到开发的build任务上。当开发人员在仿真环境上部代码的时候，我们的自动化测试用例就会被触发执行。

三，不需要经常执行的测试用例。像全量测试用例，我们没有必要一直回归执行，必竟还是有时间消耗的，有些非主要业务线也不需要时时回归。这类测试用例我们就采用人工执行，在jenkins创建一个任务，需要执行的时候人工去构建即可。

7、什么是持续集成？

- 持续集成源于极限编程(XP)，是一种软件实践，软件开发过程中集成步骤是一个漫长并且无法预测的过程。集成过程中可能会爆发大量的问题，因此集成过程需要尽可能小而多，实际上持续集成讲的是不断的去做软件的集成工作。持续集成，最简单的形式是包括一个监控版本控制(SVN等等)变化的工具。当变化被发觉时，这个工具可以自动的编译并测试你的应用。

8、自动化测试的时候是不是需要连接数据库做数据校验？

- UI自动化不需要

- 接口测试会需要

9、id,name,class,xpath, css selector这些属性，你最偏爱哪一种，为什么？

- css 、xpath 几乎所有的元素都可以定位到

10、如何去定位页面上动态加载的元素？

- 触发动态加载元素的事件，直至动态元素出现，进行定位

11、如何去定位属性动态变化的元素？

- xpath或者css通过同级、父级、子级进行定位

12、点击链接以后，selenium是否会自动等待该页面加载完毕？

- 会的

13、webdriver client的原理是什么？

- [Webdriver实现原理](http://www.cnblogs.com/timsheng/archive/2012/06/12/2546957.html)

通过研究selenium-webdriver的源码，笔者发现其实webdriver的实现原理并不高深莫测无法揣度。在这里以webdriver ruby binding的firefox-webdriver实现为例，简单介绍一下webdriver的工作原理。

当测试脚本启动firefox的时候，selenium-webdriver 会首先在新线程中启动firefox浏览器。如果测试脚本指定了firefox的profile，那么就以该profile启动，否则的话就新启1个profile，并启动firefox；

firefox一般是以-no-remote的方法启动，启动后selenium-webdriver会将firefox绑定到特定的端口，绑定完成后该firefox实例便作为webdriver的remote server存在；

客户端(也就是测试脚本)创建1个session，在该session中通过http请求向remote server发送restful的请求，remote server解析请求，完成相应操作并返回response；

客户端接受response，并分析其返回值以决定是转到第3步还是结束脚本；

这就是webdriver的工作流程，看起来很复杂实际上当了解了webdriver的实现原理后，理解上述问题应该比较简单。

webdriver是按照server – client的经典设计模式设计的。

server端就是remote server，可以是任意的浏览器。当我们的脚本启动浏览器后，该浏览器就是remote server，它的职责就是等待client发送请求并做出相应；

client端简单说来就是我们的测试代码，我们测试代码中的一些行为，比如打开浏览器，转跳到特定的url等操作是以http请求的方式发送给被 测试浏览器，也就是remote server；remote server接受请求，并执行相应操作，并在response中返回执行状态、返回值等信息；

14、webdriver的协议是什么？

-The WebDriver Wire Protocol

15、启动浏览器的时候用到的是哪个webdriver协议？

-http

16、什么是page object设计模式？

-http://www.cnblogs.com/tsbc/p/4080301.html

相似功能地方，代码基本都是一样的，界面元素换个查找方式，把原来的使用 xpath方式，改为使用 id 查找，需要对每个用例脚本都要改，虽然几个用例看不出什么工作量，但是重复findElement的代码，已经让我们感到了代码的笨重。如果某些定位发生了改变，我们就得贯穿整个测试代码进行调整元素定位，这样就会导致我们的脚本在后期，难以维护。因此通过Page Object Model 我们可以创建更加健壮代码，并减少或者消除重复的测试代码，从而也能够提高代码的可读性，减少编写脚本的工作量。Page Object Model的实现，就是通过分离测试对象和测试脚本的抽象来实现的。

17、什么是page factory设计模式？

- http://relevantcodes.com/pageobjects-and-pagefactory-design-patterns-in-selenium/

18、怎样去选择一个下拉框中的value＝xx的option？

-二次定位

19、如何在定位元素后高亮元素(以调试为目的)？

-重置元素属性，给定位的元素加背景、边框

20、什么是断言？

-断言的英文是assertion，断言检查的英文是assertion checking。

-断言是指定一个程序必须已经存在的状态的一个逻辑表达式，或者一组程序变量在程序执行期间的某个点上必须满足的条件。

21、如果你进行自动化测试方案的选型，你会选择哪种语言，java，js，python还是ruby？

-使用自己熟悉的语言

22、page object设置模式中，是否需要在page里定位的方法中加上断言？

-不需要

23、page object设计模式中，如何实现页面的跳转？

-get、click (可能有坑)

24、自动化测试用例从哪里来？

-手工用例中抽取

-可以参考自动化用例的执行策略

25、你觉得自动化测试最大的缺陷是什么？

-不稳定

-可靠性

-不易维护

-成本与收益

，

26、什么是分层测试？

-[分层自动化](http://www.cnblogs.com/fnng/p/3653793.html?utm\_source=tuicool&utm\_medium=referral)

还是自己理解吧

27、webdriver可以用来做接口测试吗？

-有难度，不推荐

————————————————

版权声明：本文为CSDN博主「weixin\_39860064」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_39860064/article/details/111755680>

python selenium面试\_selenium面试问题答案总结

weixin\_39985842 2020-12-14 23:55:27 453 收藏 4

文章标签： python selenium面试

版权

工程师小C的小店我也想开通小店

Python编程三剑客：Python编程从入门到实践第2版+快速上手第2版+极客编程（套装共3册）

作者：[美] 埃里克·马瑟斯（Eric Matthes）

出版社：人民邮电出版社

好评：100.0%

销售量：20

￥149

更多

1. 什么是Selenium？

Selenium是一个开源的web自动化测试框架，主要是基于web uI的自动化测试。现在的版本，逐步增加了对移动端的自动化测试。Selenium支持多种语言进行开发自动化测试脚本，有Java,python，C#，Javascript等等。Selenium支持跨浏览器平台测试。

2.Selenium是否支持桌面应用软件的自动化测试。

Selenium不支持桌面软件的自动化测试，Selenium是根据网页元素的属性才定位元素，而其他桌面软件自动化测试工具是根据桌面元素的位置来定位元素，当然现在也有根据桌面元素的属性来定位的。

3.Selenium是否支持用例的执行的引擎。

引擎好比就是一个发动机。Selenium是没有关于测试用例和测试套件管理和执行的模块。我们需要借助第三方单元测试框架来实现用例管理和用例的执行。例如Java中有Junit或者testNG,Python中有unittest单元测试框架。

4.Seleinum是否有读取excel文件的库

没有，这里需要用到第三方工具。例如Apache POI插件。

5.Selenium有哪些组件？

最早的有Selenium IDE,IDE只支持安装在fiefox上一个插件，支持录制自动化脚本。还有

remote RC,和Grid 和webdriver。我们一般最重要的就是使用webdriver。

6.Selenium有什么限制或者缺陷

除了基于web的软件和mobile的程序，selenium不支持桌面软件自动化测试。软件测试报告，和用例管理只能

依赖第三方插件，例如Junit/TestNG和unittest。由于它是免费的软件，所以没有供应商去提供支持和服务，有问题，只能求助selenium社区。还有一个就是，selenium入门门槛可能有点高，需要具备一定编程语言基础的才能玩转。

7.在selenium中，有哪些不同定位元素方法

ID/className/Name/LinkText/PartialLinkText/Xpath/CSS selector

8.什么是imlicitlyWait

imlicitlyWait是隐式等待，一般在查找元素的时候使用。例如，我设置一个查找元素最大时间为10秒，使用了

imlicitlyWait后，如果第一次没有找到元素，会在10秒之内不断循环去找元素，知道超过10秒，报超时错误。

9.什么是expliciteWait

这个是显式等待，就是不管如何都是要等10秒，如果你设置了10秒超时，这个是selenium2的功能

在selenium3中，我暂时没有找到这个接口。

10.什么是线程等待

有时候，我们需要强制设置线程等待，Thread.sleep(2000),driver这个实例，就是当前的线程。

11.什么是pollingEvery

这个是设置个一段时间就去做一件事，例如下面设置隔一秒就去查找元素一次。

WebDriverWait wait = new WebDriverWait(driver,30);

wait.pollingEvery(1, TimeUnit.SECONDS);

driver.findElement(By.xpath("xxxx"));

12你能解释下Selenium这个框架吗？

这个问题在面试中被问到的概率还是比较高的，同样类似的问题有，selenium的原理是什么？首先不要被这个问题吓到，我们主要围绕selenium的历史版本演化和基本的组件去展开描述就好，最后回到webdriver这个组件上面，我们基本上都是在使用webdriver提供的API。所以这个题目的最好的答案就是把图画出来，然后自己解释几句就可以。 早期Selenium1.0是有Selenium Grid，Selenium RC, Selenium IDE, Webdriver四部分组成，后来Selenium RC和Webdriver合并之后，就是Selenium2,当前我们在使用Selenium3。

Selenium Grid：它是selenium框架的一部分，主要是专门用来把测试用例并行地在不同浏览器，不同操作系统，不同机器上运行。一般我们写脚本，调试都在单机上线性地一个测试用例接着一个测试用例执行下去。如果有人问题如何提高测试用例执行效率，告诉他Selenium Grid可以实现。

Selenium IDE: 这个算Selenium里面最简单的一个组建，只支持在火狐浏览器上安装这个扩展程序，支持录制web ui脚本，然后导出不同语言的脚本，例如java c#等。这个功能算鸡肋，因为很多时候导出脚本debug的时间还不如自己代码重新写来的快。

Selenium RC: RC是remote control的缩写,主要的功能就是让你不管使用什么语言(Selenium支持的这几种语言之一)来写测试脚本，只要是这个浏览器支持java script，那么写一遍测试脚本，都能在这些不同浏览器运行脚本。

Webdriver：这个是用来替代Selenium RC，就是一个网页自动化工具，支持在不同浏览器上运行测试脚本，运行速度比Selenium RC要快很多。据说(我也记得不清楚)，webdriver最早是google内部开发的一个工具，用来捐给selenium了，变成开源了。

目前，我们做的web ui的自动化测试，大部分都是在使用webdriver提供的API来模拟手动测试过程中的一系列动作和行为。基本上通过这个方式来回答这个问题，那就没问题了。

13.你写的测试脚本能在不同浏览器上运行吗，支持跨浏览器平台吗

这里出现了跨浏览器平台的概念，就是写一个测试用例，可以在主流的几个浏览器跑起来。

是的，我写的测试用例能在IE，火狐和谷歌这三种浏览器上运行。主要是在windows平台上运行脚本，所以mac的safari浏览器暂时没有写过。主要实现这个跨浏览器的思想就是，把浏览器类型写到配置文件，代码里写if语句去判断配置文件的浏览器的类型，来决定用什么浏览器去执行测试用例。

14.一天你写多少个自动化测试用例

这个要看具体情况，完全取决于手工测试用例的实现难易程度。通常，熟练的话，写一个5到8个步骤的测试用例，差不多要半小时。时间最多花在元素定位和报错debug上面，例如在POM思想的框架中，某一些元素定位和方法是复用的，可能会更快一些。所以，一天，大概能完成15-30个自动化测试用例。

15.什么是POM，为什么要使用它

POM是Page Object Model的简称，它是一种设计思想，而不是框架。大概的意思是，把一个一个页面，当做一个对象，页面的元素和元素之间操作方法就是页面对象的属性和行为，所以自然而然就用了类的思想来组织我们的页面。一般一个页面写一个类文件，这个类文件包含该页面的元素定位和业务操作方法。

为了我们测试用例写的简单，清晰，我们很多时候在页面对象会封装很多业务操作方法，测试脚本只需要调用相关方法就可以。

还有一个可能和这个问题相关的面试题，如果页面元素经常发生需求变化，你是如何做，答案就是采用POM思想。好处就是只要该一个页面，我就去修改这个页面对象的元素定位和相关方法，脚本不需要修改。

16.在你做自动化过程中，遇到了什么问题吗？举例下

这个问题，不管是自动化还是任何工作，都会被问到。主要想知道你是如何解决问题的，从而推断你问题分析和解决的能力。

当然有遇到问题和挑战，主要有以下几点：

频繁地变更UI，经常要修改页面对象里面代码

运行用例报错和处理，例如元素不可见，元素找不到这样异常

测试脚本复用，尽可能多代码复用

一些新框架产生的页面元素定位问题，例如ck编辑器，动态表格等

这个遇到的难点完全取决写脚本人的代码能力。回答三个左右就差不多，记得既然抛出了难点问题，一定要记得处理这个问题的方法。

17.举例一下你遇到过那些异常，在selenium自动化测试过程中

通过这个问题，大概知道你写过多少脚本。写脚本过程最常见的异常就是，这个元素无法找到。常见的selenium有以下这些：

1. ElementNotSelectableException ：元素不能选择异常

2. ElementNotVisibleException ：元素不可见异常

3. NoSuchAttributeException ：没有这样属性异常

4. NoSuchElementException：没有该元素异常

5. NoSuchFrameException ：没有该frame异常

6. TimeoutException ： 超时异常

7. Element not visible at this point  ：在当前点元素不可见

18. 如何处理alert弹窗

我们常见的alert弹窗有两种：基于windows弹窗和基于web页面弹窗

我们知道，webdriver是能够处理alert弹窗的，Selenium提供了Alert这个接口。相关操作代码如下：

// 切换到Alert

Alert alert = driver.switchTo().alert();

// 点击弹窗上确定按钮

alert.accept();

// 点击弹窗的取消按钮

alert.dismiss()

// 获取弹窗上线上的文本文字内容

alert.getText();

// 有些弹窗还支持文本输入，这个可以把要输入字符通过sendkeys方法输入

alert.sendkeys();

19. 在selenium中如何处理多窗口？

这个多窗口之间跳转处理，在实际selenium自动化测试经常遇到。就是，你点击一个链接，这个链接会在一个新的tab打开，然后你接下来要查找元素在新tab打开的页面，所以这里需要用到swithTo方法。

需要获取当前浏览器多窗口句柄，然后根据判断跳转新句柄还是旧句柄，具体代码可以参考我博客文章：http://blog.csdn.net/u011541946/article/details/73611301

20. 你查找元素遇到过在Frame里面吗?你是如何处理Frame里面元素定位的？

有时候我们知道元素定位表达式没有问题，但是还是提示no such element，那么我们就需要考虑这个元素是否在frame中。如果在，我们就需要从topwindow，通过swithcTo.Frame()方法来切换到目标frame中，可以通过frame的name和id和索引三种方法来定位frame。

21. 怎么验证勾选框是enable/disabled/ checked/Unchecked/ displayed/ not displayed？

通过以下方法来验证元素是enable 还是disable

boolean enabled = driver.findElement(By.xpath("元素定位表达式")).isEnabled();

通过以下方法来验证元素是select/check

boolean checked = driver.findElement(By.xpath("元素定位表达式")).isSelected();

通过以下方法来验证元素是dispalyed还是not display

boolean displayed = driver.findElement(By.xpath("元素定位表达式")).isDisplayed();

22. 如何处理下拉菜单？

通常我们也可以通过Click方法来点击下拉菜单里面的元素，还有一种方法，在Selenium中有一个类叫Select，支持这种下拉菜单交互的操作。

基本使用语法是这样的：

Select Se=new Select(element);

Se.selectByIndex(index);

Se.selectByvalue(value);

Se.selectByVisibleText(text);

23. 在日历这种web 表单你是如何处理的?

首先要分析当前网页试用日历插件的前端代码，看看能不能通过元素定位，点击日期实现，如果不能，可能需要借助javascript。还有些日历控件一个文本输入框，可以直接sendKeys()方法来实现传入一个时间的数据。

24. 关闭浏览器中quit和close的区别

简单来说，两个都可以实现退出浏览器session功能，close是关闭你当前聚焦的tab页面，而quit是关闭全部浏览器tab页面，并退出浏览器session。知道这两个区别，我们就知道quit一般用在结束测试之前的操作，close用在执行用例过程中关闭某一个页面的操作。

25. 什么是页面加载超时

Selenium中有一个 Page Load wait的方法，有时候，我们执行脚本的速度太快，但是网页程序还有一部分页面没有完全加载出来，就会遇到元素不可见或者元素找不到的异常。为了解决问题，让脚本流畅的运行，我们可以通过设置页面加载超时时间。具体代码是这个：driver.manage().timeouts().pageLoadTimeout(10,TimeUnit.SECONDS);

这行作用就是，如果页面加载超过10秒还没有完成，就抛出页面加载超时的异常。

26.什么是JavaScript Executor，你什么时候会用到这个？

JavaScript Executor是一个接口，给driver对象提供一个执行javaScript并访问和修改前端元素属性和值。

还是有比较多的场景，我们可能或者需要借助javaScript来实现：

1.元素通过现有定位表达式不能够实现点击

2.前端页面试用了ck-editor这个插件

3.处理时间日期插件(可能)

4.生成一个alert弹窗

5.拖拽滚动条

基本语法：

JavascriptExecutor js =(JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

27.在Selenium中如何实现截图，如何实现用例执行失败才截图

在Selenium中提供了一个TakeScreenShot这么一个接口，这个接口提供了一个getScreenshotAs()方法可以实现全屏截图。然后我们通过java中的FileUtils来实现把这个截图拷贝到保存截图的路径。

代码举例：

File src=((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

try {

// 拷贝到我们实际保存图片的路径

FileUtils.copyFile(src,new File("C:/selenium/error.png"));

}

catch (IOException e)

{

System.out.println(e.getMessage());

}

如果要实现执行用例发现失败就自动截图，那么我们需要把这个截图方法进行封装。然后在测试代码中的catch代码块去调用这个截图方法。这个我们在POM的框架中一般是把截图方法封装到BasePage这个文件中。

28.在Selenium中如何实现拖拽滚动条？

在Selenium中通过元素定位会自动帮你拖拽到对应位置，所以是没有自带的scoll方法。但是这个是有限制，例如当前页面高度太长，默认是页上半部分，你定位的元素在页尾，这个时候可能就会报元素不可见的异常。我们就需要利用javaScript来实现拖拽页面滚动条。

我们一般可以两个方法去拖拽，一个是根据拖拽的坐标(像素单位)，另外一个是根据拖拽到一个参考元素附件。

代码举例(根据元素坐标拖拽)：

JavascriptExecutor jse= (JavascriptExecutor)driver;

jse.executeScript("window.scrollBy(0,250)", "");

29.如何实现文件上传？

我们在web页面实现文件上传过程中，可以直接把文件在磁盘完整路径，通过sendKeys方法实现上传。如果这种方法不能实现上传，我们就可能需要借助第三方工具，我用过一个第三方工具叫autoIT.

还有一个方法是利用robot类：

http://blog.csdn.net/u011541946/article/details/74332938

30.如何处理“不受信任的证书”的问题？

例如，在登录12306网站的时候，如果你没有下载和安装过这个网站的根证书，那么你就会遇到打开12306网站提示证书不受信任的拦截页面。

下面举例火狐和谷歌上处理这个问题的基本代码

火狐：

// 创建firefoxprofile

FirefoxProfile profile=new FirefoxProfile();

// 点击继续浏览不安全的网站

profile.setAcceptUntrustedCertificates(true);

// 使用带条件的profile去创建一个driver对象

WebDriver driver=new FirefoxDriver(profile);

Chrome：

// 创建类DesiredCapabilities的对象

DesiredCapabilities cap=DesiredCapabilities.chrome();

// 设置ACCEPT\_SSL\_CERTS 变量值为true

cap.setCapability(CapabilityType.ACCEPT\_SSL\_CERTS, true);

// 新建一个带capability的chromedriver对象

WebDriver driver=new ChromeDriver(cap);

31.findElement 和 FindElements有什么区别？

首先，两个都是查找元素，都支持八大元素定位方法。findElement()得到的只有一个元素，如果根据提供的元素定位方式找不到，会报noSuchElement异常。

findElements()返回的是一组元素，所以我们需要根据能够找到一组元素的表达式去定位，返回一组元素我们可以放在集合里，这样我们就可以使用集合里面的迭代方法去遍历元素，拿到元素去做其他操作。

32.在执行脚本过程，如何实现当前元素高亮显示？

这个其实就是利用javaScript去修改当前元素的背景颜色来到达高亮显示的效果，

33.如何获取页面标题，悬浮文本和错误文本，并验证？

标题，我们可以通过driver.getTitle()方法来得到一个字符串，然后使用字符串的containts方法或者equals方法去进行断言。

悬浮文本(tooltip)，一般是利用Actions类，然后鼠标悬停方法，然后通过getText()方法来得到这个tooltip字符串。

错误信息，直接把这个错误字段先进行定位，然后通过getText()方法拿到错误文本，主要的断言有包含，相等，不相等，不包含，以什么开头等。

34.在selenium自动化测试中，你一般完成什么类型的测试？

主要是冒烟测试和回归测试。回归测试主要写一些功能稳定，容易实现的场景，通过自动化手段去实现，节约测试时间。

35.你是如何管理你的测试用例并执行？

写用例和管理并执行用例，我们都需要借助单元测试框架来实现，如果是Java语言一般有junit和TestNG，如果是python，常见的有unittest。

就你实际情况，说一下。例如我使用TestNG比较多，需要配置testng.xml文件来实现测试用例的执行。有时候需要配置多个testng.xml去实现不同的任务场景。再展开，可能问你一下testng框架的知识点。例如，方法依赖，用例执行优先级，数据源驱动等。

36.关于自动化测试报告生成？

我个人一般用TestNG原生的测试报告，也有第三方叫reportNG的插件，不过我没有实际使用过。

Python下报告生成一般使用HTMLTestRunner.py

37. 了解或者使用过框架不？

类似的问题还有，你知道那些自动化测试框。

我们知道POM自动化测试框，还有关键字驱动框架。当然还有数据驱动框架，最近几年出的行为驱动框架。

38. 这个框架是你自己写的吗？

我自己设计过POM的框架和关键字驱动框架。

39.能不能介绍下你的框架

把你自己POM的框架从上往下，树形结构画图出来，然后讲下有哪些层，哪些包，包下

————————————————

版权声明：本文为CSDN博主「weixin\_39985842」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_39985842/article/details/111424529>

1.你能解释下Selenium这个框架吗？

    这个问题在面试中被问到的概率还是比较高的，同样类似的问题有，selenium的原理是什么？首先不要被这个问题吓到，我们主要围绕selenium的历史版本演化和基本的组件去展开描述就好，最后回到webdriver这个组件上面，我们基本上都是在使用webdriver提供的API。所以这个题目的最好的答案就是把图画出来，然后自己解释几句就可以。

    早期Selenium1.0是有Selenium Grid，Selenium RC, Selenium IDE, Webdriver四部分组成，后来Selenium RC和Webdriver合并之后，就是Selenium2,当前我们在使用Selenium3

Selenium Grid：它是selenium框架的一部分，主要是专门用来把测试用例并行地在不同浏览器，不同操作系统，不同机器上运行。一般我们写脚本，调试都在单机上线性地一个测试用例接着一个测试用例执行下去。如果有人问题如何提高测试用例执行效率，告诉他Selenium Grid可以实现。

Selenium IDE: 这个算Selenium里面最简单的一个组建，只支持在火狐浏览器上安装这个扩展程序，支持录制web ui脚本，然后导出不同语言的脚本，例如java c#等。这个功能算鸡肋，因为很多时候导出脚本debug的时间还不如自己代码重新写来的快。

Selenium RC: RC是remote control的缩写,主要的功能就是让你不管使用什么语言（Selenium支持的这几种语言之一）来写测试脚本，只要是这个浏览器支持java script，那么写一遍测试脚本，都能在这些不同浏览器运行脚本。

Webdriver：这个是用来替代Selenium RC，就是一个网页自动化工具，支持在不同浏览器上运行测试脚本，运行速度比Selenium RC要快很多。据说（我也记得不清楚），webdriver最早是google内部开发的一个工具，用来捐给selenium了，变成开源了。

    目前，我们做的web ui的自动化测试，大部分都是在使用webdriver提供的API来模拟手动测试过程中的一系列动作和行为。基本上通过这个方式来回答这个问题，那就没问题了。

2.你写的测试脚本能在不同浏览器上运行吗，支持跨浏览器平台吗

这里出现了跨浏览器平台的概念，就是写一个测试用例，可以在主流的几个浏览器跑起来。

    是的，我写的测试用例能在IE，火狐和谷歌这三种浏览器上运行。主要是在windows平台上运行脚本，所以mac的safari浏览器暂时没有写过。主要实现这个跨浏览器的思想就是，把浏览器类型写到配置文件，代码里写if语句去判断配置文件的浏览器的类型，来决定用什么浏览器去执行测试用例。

这个过程，具体看我的博客文章：http://blog.csdn.net/u011541946/article/details/76598441

3.一天你写多少个自动化测试用例

    这个要看具体情况，完全取决于手工测试用例的实现难易程度。通常，熟练的话，写一个5到8个步骤的测试用例，差不多要半小时。时间最多花在元素定位和报错debug上面，例如在POM思想的框架中，某一些元素定位和方法是复用的，可能会更快一些。所以，一天，大概能完成15-30个自动化测试用例。

4.什么是POM，为什么要使用它

    POM是Page Object Model的简称，它是一种设计思想，而不是框架。大概的意思是，把一个一个页面，当做一个对象，页面的元素和元素之间操作方法就是页面对象的属性和行为，所以自然而然就用了类的思想来组织我们的页面。一般一个页面写一个类文件，这个类文件包含该页面的元素定位和业务操作方法。

为了我们测试用例写的简单，清晰，我们很多时候在页面对象会封装很多业务操作方法，测试脚本只需要调用相关方法就可以。

还有一个可能和这个问题相关的面试题，如果页面元素经常发生需求变化，你是如何做，答案就是采用POM思想。好处就是只要该一个页面，我就去修改这个页面对象的元素定位和相关方法，脚本不需要修改。

具体可以看文章：http://blog.csdn.net/u011541946/article/details/75389004

5.在你做自动化过程中，遇到了什么问题吗？举例下

    这个问题，不管是自动化还是任何工作，都会被问到。主要想知道你是如何解决问题的，从而推断你问题分析和解决的能力。

当然有遇到问题和挑战，主要有以下几点：

频繁地变更UI，经常要修改页面对象里面代码

运行用例报错和处理，例如元素不可见，元素找不到这样异常

测试脚本复用，尽可能多代码复用

一些新框架产生的页面元素定位问题，例如ck编辑器，动态表格等

    这个遇到的难点完全取决写脚本人的代码能力。回答三个左右就差不多，记得既然抛出了难点问题，一定要记得处理这个问题的方法。

————————————————

版权声明：本文为CSDN博主「Anthony\_tester」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/u011541946/article/details/79400486>

    软件测试培训也是一种提升技术及迅速转行的辅助手段，不管在哪学你都必须先去试听，看看讲课的风格及氛围，是否通俗易懂，课件是否老化，工具是否过时，课程内容及项目是否具有实战性，主要就是了解项目实战周期是多久，工具是否属现在主流，还有你自己喜不喜欢讲师的讲课风格，如果不喜欢讲师的上课风格也会对你的学习进度造成一定的影响，今天给大家分享Selenium的面试题。希望在学习的小伙伴都能找到更好的心仪工作。

问题1：Selenium是什么，流行的版本有哪些？

Selenium是基于Web的最流行的UI自动化测试工具。它提供了一组支持多种平台的公开API（例如Linux，Windows，Mac OS X等）。此外，像Google Chrome，Mozilla Firefox，Internet Explorer和Safari等所有现代浏览器都可以用来运行Selenium测试。它也涵盖了Android平台，其中Appium是实现Selenium Webdriver界面的工具，用于移动自动化。

值得注意的是，除了许多后来的小型版本之外，硒还有三个主要版本：

Selenium 1.0或Selenium RC，于2004年初发布，提供了一个使用服务器与浏览器交换命令和响应的API集。

Selenium 2.0或Selenium Webdriver，在2011年中推出，并在Selenium功能中引入了一系列重大改进。这些新的API完全取代了服务器组件，并与目标浏览器本地交互。

Selenium 3.0，这个版本是在2016年末发布的大版本。它带来的主要变化是引入Webdriver API的W3C规范，用于浏览器自动化。也就是说，每个主要的浏览器都会有自己的Webdriver API来实现功能。

问题2：你如何从命令行启动Selenium RC？

// 简单的启动Selenium RC的方法是

java -jar selenium-server.jar

// 在浏览器中运行一套Selenese脚本

java -jar selenium-server.jar –htmlSuite

问题3：在我的机器端口4444不是免费的。我怎样才能使用另一个端口？

//你可以在运行selenium服务器时指定端口为 -

Java -jar selenium-server.jar -port 5555

问题4：什么是Selenium Server，它与Selenium Hub有什么不同？

Selenium Server是使用单个服务器作为测试节点的一个独立的应用程序。 Selenium hub代理一个或多个Selenium的节点实例。一个hub 和多个node被称为Selenium grid。运行SeleniumServer与在同一主机上用一个hub和单个节点创建de Selenium grid类似。

问题5：你如何从Selenium连接到数据库？

Selenium是一个Web UI自动化工具。它不提供任何API来建立数据库连接。这取决于你使用Selenium进行自动化的编程语言。在下面的例子中，我们假设正在使用Java。

一个Connection对象表示与数据库的连接。当我们使用连接方法连接到一个数据库时，我们创建了一个连接对象，它代表了与数据库的连接。单个数据库可能有一个连接或多个连接，还可能有多个连接到不同的数据库上。

我们可以使用Connection对象来做以下事情：

创建用于执行SQL语句的Statement，PreparedStatement和CallableStatement对象。

可以帮助我们提交或回滚一个JDBC事务。

如果你想知道连接到的数据库或数据源信息，Connection对象通过使用DatabaseMetaData就可以收集有关数据库或数据源的信息。

可以帮助我们关闭数据源。Connection.isClosed() 方法只有在调用了Connection.close()时才返回true 。此方法用于关闭所有连接。

首先我们需要通过使用DriverManager.getConnection()方法，建立与数据库的连接。这个方法接受一个包含URL的字符串。DriverManager类尝试查找可以连接到由字符串URL表示的数据库的驱动程序。每当调用getConnection()方法时，DriverManager类都会检查可以连接到URL中指定的数据库的所有已注册的Driver类的列表。

句法：

String url = "jdbc: odbc:makeConnection";

Connection con = DriverManager.getConnection(url,"userID", "password");

问题6：Selenium RC中定位器方法有哪些？

00001.

ID

00002.

00003.

Name

00004.

00005.

CSS (Cascade Stylesheet)

00006.

00007.

XPATH (Relative XPath and Absolute XPath)

00008.

00009.

Dom

00010.

问题7：你如何验证多个页面上存在的一个对象？

可以使用下面的Selenium命令来检查：

assertTrue(selenium.isElementPresent(locator));

问题8：XPath中使用单斜杠和双斜杠有什么区别？

如果XPath是从文档节点开始，它将允许创建“绝对”路径表达式。

例如 “/html / body / p”匹配所有的段落元素。

如果XPath在文档中的任意位置开始进行选择匹配，那么它将允许创建“相对”路径表达式。

例如“// p”匹配所有的段落元素。

问题9：如何编写Selenium IDE / RC的用户扩展？

用户扩展（UX）存储在Selenium IDE或Selenium RC用来激活扩展的单独文件中。它包含用JavaScript编写的函数定义。

因为Selenium的核心是用JavaScript开发的，所以要符合原语言的标准规则来创建扩展。要创建一个扩展，我们必须用下面的设计格式来编写函数。

// 样例

Selenium.prototype.doFunctionName = function(){

}

函数名称前面的“do”告诉Selenium这个函数可以被调用为一个步骤命令，而不是作为内部函数或私有函数被调用。

问题10：如何在页面加载成功后验证元素的存在？

它可以通过下面的代码行来实现。

只需一点时间（以秒为单位）来检查元素，如下所示：

public void waitForElementPresent(String element,int timeout) throws Exception {

for (int second = 0;; second++) {

if (second >= timeout)

fail("Timeout. Unable to find the Specifiedelement" + element);

try {

if (selenium.isElementPresent(element))

break;

} catch (Exception e) {

}

Thread.sleep(1000);

}

}

如果你不想再体验一次自学时找不到资料，没人解答问题，坚持几天便放弃的感受的话，可以加我们的软件测试学习交流群：705673780，里面有各种软件测试资料和技术交流~

————————————————

版权声明：本文为CSDN博主「a575223」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/a575223/article/details/81044044>

**elenium 常见面试题以及答案（Java版）**



[喜欢安静的程序猿](https://blog.csdn.net/weixin_39363245) 2019-07-18 06:14:59  229  收藏 2

**1.怎么 判断元素是否存在？**

判断元素是否存在和是否出现不同， 判断是否存在意味着如果这个元素压根就不存在， 就会抛出NoSuchElementException

这样就可以使用try catch，如果catch到NoSuchElementException 就返回false

**2.如何判断元素是否出现？**

判断元素是否出现，存在两种情况，一种是该元素压根就没有，自然不会出现；另外一种是有这样的元素，但是是hidden状态

可以通过先判断是否存在，如果不存在返回false；如果存在再去判断是否displayed

**3. 怎样选择下拉菜单的元素**

 下拉菜单分两种，一种是直接使用select标签的，这种情况可以直接使用selenium API

参考：http://www.cnblogs.com/tobecrazy/p/4570494.html

复制代码

WebElement selector = driver.findElement(By.id("Selector"));

Select select = new Select(selector);

选择select的option有以下三种方法

selectByIndex(int index) 通过index

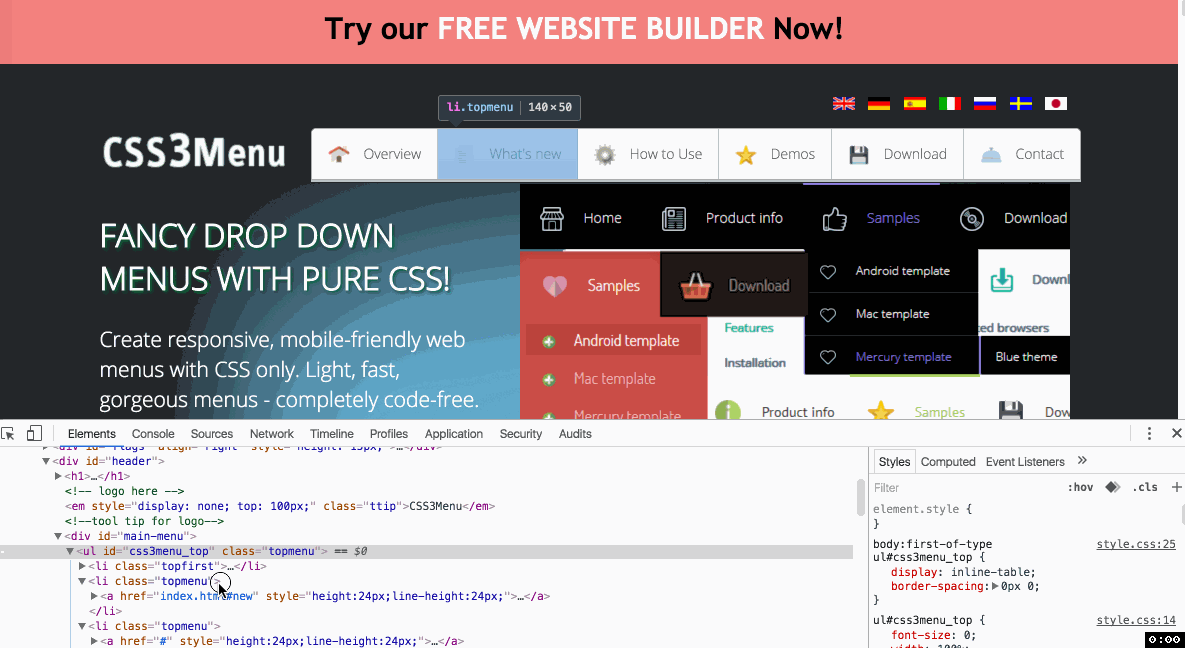
selectByVisibleText(String text) 通过匹配到的可见字符

selectByValue(String value) 通过匹配到标签里的value

复制代码

第二种下拉菜单不是通过select实现的，可以通过JS进行操作

类似这样的：http://css3menu.com/ (关于怎么使用DevTools 请自行百度，不解释)



那么这样的菜单该怎么去选取？

可以收工演示一下，第一步鼠标移动到how to use，此时菜单出现；第二步，点击Technical Question

要实现第一步，使用selenium 的Action clickAndHold，接着就可findByElement进行操作

复制代码

WebElement menu = driver.findElement(By.xpath("//span[.='How to Use']"));

Actions action = new Actions(driver);

action.clickAndHold(menu).build().perform();

WebElement technicalQuestion = driver.findElement(By.xpath(

"//ul[@id='css3menu\_top']/li[position()=3]/div[@class='submenu']/div[@class='column']//a[contains(text(),'Technical')]"));

technicalQuestion.click();

复制代码

**4. selenium 有几种定位方式，你最常用哪种， 为什么？**

selenium有八种定位方式，和name有关的3个ByName，ByClassName，ByTagName

                                   和link有关的2个ByLinkText，ByPartialLinkText

                                   和id有关的1个ById

                                  剩下两个全能的ByXpath和ByCssSelector

我最常用的事ByXpath（或CssSelector）因为很多情况下，html标签的属性不够规范，无法通过单一的属性定位，这个时候就只能使用xpath可以去重实现定位唯一element

事实上定位最快的应当属于ById，因为id是唯一的，然而大多数开发并没有设置id

**5.去哪网面试题Java实现**

1. 一、 UI自动化测试
2. 1、 Qunar机票搜索场景
3. 1) 访问Qunar机票首页http://flight.qunar.com，选择“单程”，输入出发、到达城市，选择today+7日后的日期，点“搜索”，跳转到机票单程搜索列表页。
4. 2) 在列表页停留1分钟，至到页面上出现“搜索结束”。
5. 3) 如果出现航班列表，对于出现“每段航班均需缴纳税费”的行随机点选“订票”按钮，在展开的列表中会出现“第一程”、 “第二程”；对于没有出现“每段航班均需缴纳税费”的行随机点选“订票”按钮，在展开的列表底部中会出现“报价范围”
6. 4) 如果不出现航班列表，则页面会出现“该航线当前无可售航班”

参考我的blog， <http://www.cnblogs.com/tobecrazy/p/4752684.html>

**6. 如何去定位页面上动态加载的元素？**

    触发动态事件事件，进而findElemnt

    如果是动态菜单，需要一级一级find

**7.如何去定位属性动态变化的元素？**

　 属性动态变化是指该element没有固定的属性值，所以只能通过相对位置定位

    比如通过xpath的轴， parent／following－sibling／precent－sibling等

    另外也可以尝试findbyelements遍历

**8.怎么提高selenium脚本的自动化执行效率？**

* 优化测试用例，尽可不使用 sleep，减少使用ImplicitlyWait  
  ，而使用selenium的wait／FluentWait，这样可以优化等待时间
* 使用selenium grid，通过testng实现并发执行
* 针对一些不稳定的动态控件通过JS实现操作
* 重载testng的listener实现retry机制，提高测试用例成功率
* 减少使用IE的driver，IE执行效率太低！！！

**9. webdriver 的原理是什么？**

参考：http://www.cnblogs.com/tobecrazy/p/5034408.html

**10. selenium中如何保证操作元素的成功率？也就是说如何保证我点击的元素一定是可以点击的？**

参考：<http://www.cnblogs.com/tobecrazy/p/4817946.html>

* 通过封装find方法实现waitforEmelentPresent，这样在对元素进行操作之前保证元素被找到，进而提高成功率
* 在对元素操作之前，比如click，如果该元素未display（非hidden），就需要先滚动到该元素，然后进行click操作

　　为啥使用滚动？ 因为如果页面没有完全显示，element如果是在下拉之后才能显示出来，只能先滚动到该元素才能进行click，否则是不能click操作

|  |  |
| --- | --- |
| 1  2  3 | JavascriptExecutor js=(JavascriptExecutor)driver;          // roll down and keep the element to the center of browser          js.executeScript("arguments[0].scrollIntoViewIfNeeded(true);", download); |

**11. 什么PO模式，什么是page factory？**

    网上有很多答案，都不全面

    PO模式是page object model的缩写，顾名思义， 是一种设计模式，实现脚本的page和真实的网站页面Map起来，一一对应起来。这样能测试框架更容易维护。 比如一个登陆页面，使用PO模式后，会创建一个LoginPage的class，该class会定义用户名输入框，密码输入框，登陆按钮的webElenent

针对相应的Element实现相应的方法，输入框是用来输入的，就需要创建输入用户名和输入密码的方法，这样就和真实的页面一致，所以这样的设计理念就是PO模式。 而PageFactory隶属PO模式，是用来初始化每个PO模式实现的Page Class，初始化对象库。

# selenium 总结



[xianjie0318](https://xianjie0318.blog.csdn.net/) 2018-06-05 11:33:15  423  收藏 3

1. WebDriver原理  
  
webDriver是按照client/server模式设计，client就是我们的测试代码，发送请求，server就是打开的浏览器来打开client发出的请求并做出响应。  
  
具体的工作流程：  
  
·webdriver打开浏览器并绑定到指定端口。启动的浏览器作为remote server.  
  
·client通过commandExecuter发送httpRequest给remote server的的侦听端口（the      webdriver wire control通信协议）  
  
·Remote server根据原生的浏览器组件来转化为浏览器的native调用。  
  
备注：启动浏览器的时候，使用的是http: hyper test transfer protocol  
  
   
  
2. Selenium 特点和组成  
  
Selenium是一个针对web应用的开源测试框架，它的测试用例可以用html table或者html 代码或者编程语言进行开发，而且他能在几乎所有的现在的浏览器上执行。Selenium可以被部署到windows.linux.mac平台上，支持的语言有java/python/ruby/.net/perl  
  
特点：  
  
·支持录制回放  
  
·多种方式对页面元素进行定位并操作  
  
·能够及进行debug和设置断点  
  
·能够把测试脚本保存成html/ruby或者其他语言  
  
·支持user-extensions.js形式的用户扩展  
  
·能够进行页面的断言  
  
1) Selenium IDE: 是firefox的插件，可以录制/回放/测试脚本  
  
2) Selenium RC：支持程序语言（java/C#/python/ruby）编写测试用例; selenium和其他测试框架集成，比如java环境下将selenium和Junit集成，利用selenium来书写测试用例，用Junit来实现测试用例的自动化运行。  
  
3) Selenium Core：是selenium的核心，是有js和html文件组成，它是selenium IDE和selenium RC的核心引擎。  
  
4) Selenium Grid：是selenium的扩展，可以将一批selenium脚本分配到不同的测试机上同步运行。即selenium grid可以在不同的主机上建立主节点(hub)和分支节点(node), 可以使主节点上的测试用例在不同的分支节点上运行。对于不同的节点来说，可以搭建不同的测试环境（操作系统/浏览器）。  
  
   
  
3. Selenium内部运行机制  
  
   
  
4. 如何提高自动化脚本稳定性  
  
找原因  
  
1)网速原因，增加时间等待  
  
2)函数原因，尽量少使用容易冲突的函数  
  
3)配置testNG实现多线程，在编写测试用例的时候，一定要实现松耦合，在服务器允许的情况下尽量设置多线程运行，提高执行速度。  
  
   
  
5. 高质量自动化脚本特点  
  
业务和代码分离，封装型好。  
  
自动化用例耦合性低，独立性强，易于扩展维护  
  
   
  
6. 自动化测试缺陷  
  
不稳定  
  
可靠性不强  
  
不易维护  
  
成本与收益  
  
   
  
7. 自动化用例的执行策略  
  
根据自动化执行的目的来决定  
  
（1）自动化测试用例执行用来监控，可以设置用例定时执行；  
  
（2）用于回归测试，可以把测试用例设置成触发式执行；  
  
（3）不需要经常执行的测试用例，可以由人工执行。  
  
   
  
8. 提高selenium脚本的执行速度方式  
  
（1）减少操作步骤，减少不必要的操作；  
  
（2）中断页面加载，如果页面加载内容过多并且加载的内容不影响我们测试，可以设置超时时间，中断页面加载；  
  
（3）设置等待时间的时候，可以sleep()固定的时间，检测到元素出现后中断等待也可以提高速度；  
  
（4） 配置testNG实现多线程，在编写测试用例的时候，一定要实现松耦合，在服务器允许的情况下尽量设置多线程运行，提高执行速度。  
  
   
  
9. 元素定位  
  
1) Id定位  
  
find\_element\_by\_id(‘kw’)   //通过id属性定位  
  
2) Name定位  
  
find\_element\_by\_name(‘wd’)    //通过名字进行定位，名字一般不唯一，一般不能        通    过其定位  
  
3) ClassName定位  
  
find\_element\_by\_class\_name(‘cn’)   //通过类名进行定位  
  
4) TagName定位  
  
find\_element\_by\_tag\_name(‘input’)  //标签，一般用于一类元素的定位  
  
5) LinkText定位  
  
find\_element\_by\_link\_text(‘新闻’)  //文本链接定位  
  
6) PartialLinkText定位  
  
find\_element\_by\_partial\_link\_text(‘一个很长的’) //文本链接的部分文字  
  
7) 绝对路径定位  
  
   find\_element\_by\_xpath(‘/html/body/div/div[2]/form/span/input[2]’) //绝对地址定位，                 利用层级定位  
  
   find\_element\_by\_xpath(‘//input[@name=’n’ and id=’kw’]’) //当前页面某个目录下                         name为n，id为kw的input标签，利用元素属性进行定位  
  
   find\_element\_by\_xpath(‘//\*[@id=kw]’) //其中\*表示任何属性，表示id为kw的任何属                 性的元素  
  
   find\_element\_by\_xpath(‘//span[@class=’c’]/input’) //层级和属性结合进行定位  
  
   find\_element\_by\_xpath(‘//a[contains(href , ’test’)]’)  
  
8) 相对路径定位  
  
相对路径定位一般比绝对路径定位快。  
  
       find\_element\_by\_css\_selector(‘.class1’) // 类名定位 .class  
  
   find\_element\_by\_css\_selector(‘#id’) //id定位 #id  
  
   find\_element\_by\_css\_selector(‘input’) // 标签名定位  
  
   find\_element\_by\_css\_selector(‘[autocomplete=off]’) //属性定位  
  
   find\_element\_by\_css\_selector(‘span>input’) //标签名为span的input子元素  
  
9) 定位一组元素  
  
Checkboxes = find\_elements\_by\_tag\_name(‘checkbox’)  
  
for checkbox in Checkboxes:  
  
checkbox.click()  # 通过for循环全选所有的checkbox  
  
Checkboxes.pop().click()  #取消勾选一些checkbox，pop(-1)=pop()表示一组元素的最后一个；pop(0)      表示一组元素的第一个，pop(1)表示一组元素的第二个，以此类推  
  
10) 父子/兄弟/相邻节点的定位  
  
·父->子  
  
Find\_element\_by\_id(‘parent’).find\_element\_by\_tag\_name(‘input’) 串联寻找  
  
Find\_element\_by\_xpath(‘//input/div[@name=’n’]/span/input[1]’)  xpath的层级  
  
Find\_element\_by\_css\_selector(‘div#b>input’) css selector的父子关系  
  
·子->父  
  
Find\_element\_by\_xpath(‘//div[@id=’c’]/../..’) 其中//div[@id=’c’]的父亲的父亲  
  
·哥哥->弟弟  
  
Find\_element\_by\_xpath(‘//div[@id=’c’’/../div[1]) 找到弟弟后找父亲然后找哥哥  
  
Find\_element\_by\_css\_selector(‘div#d+div’) 表示id=d的div后面紧跟着的div  
  
Find\_element\_by\_css\_selector(‘div#d~div’) 表示id=d的div后面同一级所有div  
  
11) 二次定位  
  
Element1 = driver.find\_element\_by\_id(‘ID1’)  
  
Element2 = Element1.find\_element\_by\_id(‘ID2’)  
  
12) display: none 和hidden的区别  
  
共同点：把网页中的某个元素隐藏起来；他们在selenium中都是定位不到的。  
  
区别：display:none不为隐藏的对象保留其物理空间，该对象在这个页面上彻底消失，看不到/摸不到；hidden使对象在网页上不可见，但该对象在网页中依然占有空间，看不到/摸得到。  
  
备注：isElementPresent()来判断元素是否存在。  
  
   
  
10. 常见控件使用  
  
1) link/button:  
  
element.click()  
  
注：ajax不能使用clickAndWait()函数，因为cickAndWait函数会在click之后等待页面重新加载完成，而ajax是部分刷新，所以这个页面不能重新加载完成。  
  
2) Textbox:  
  
element.send\_keys(‘test’)  
  
   
  
3) Upload  
  
Driver.find\_element\_by\_id(‘id’).send\_keys(‘D\test.txt’)  
  
   
  
4) Mouse Event:  
  
e1=driver.find\_element\_by\_id(‘kw’)  
  
ActionChains(driver).double\_click()/.context\_click()/.drag\_and\_drop()/.move\_to\_eleme        nt()/.perform()   # 双击/右击/拖动/悬停/执行actionChains中的行为  
  
   
  
5) Dropdown:  
  
·<Select>标签的下拉菜单  
  
from selenium.webdriver.support.ui import Select  
  
Select(driver.find\_element\_by\_id(‘gender’)).select\_by\_index(1)  
  
Select(driver.find\_element\_by\_id(‘gender’)).select\_by\_value(“2”)  
  
Select(driver.find\_element\_by\_id(‘gender’)).select\_by\_visible\_text(“Male”)  
  
·非<Select>标签的下拉菜单  
  
Dropdown1 = driver.find\_element\_by\_id(‘id’) #先定位到dropdown  
  
Dropdown1.find\_element\_by\_id(“li2\_input\_2”) #再定位到dropdown中的值  
  
   
  
6) Alert:  
  
driver.switch\_to.alert.accept()//接受  
  
driver.switch\_to.alert.dismiss() //取消  
  
Message=driver.switch\_to.alert.text  //获取弹窗的文本消息  
  
driver.switch\_to.alert.send\_keys(‘hello’) //输入值,这个在alert和confirm中不可用  
  
   
  
7) Window  
  
Driver.refresh() 刷新  
  
Driver.back() 后退  
  
Driver.forward() 前进  
  
Driver.maximize\_window() 最大化  
  
Driver.set\_window\_size(100,200) 设置窗口大小  
  
Driver.switch\_to.window(searchwindow)  
  
   
  
8) Frame  
  
Driver.switch\_to.frame(ReferenceFrame)  
  
Driver.switch\_to.parent\_frame()  # frame需要一级一级切  
  
Driver.switch\_to.default\_content()  
  
   
  
11. 等待  
  
1) 显式等待  
  
等待某个条件成立时继续执行，每隔一段时间进行检测，超出最大时间    则抛出异常（程序每隔X秒看一眼，如果条件成立，执行下一步，否则等待，超出最大    时间，抛出异常）  
  
From selenium.webdriver.common.by import By  
  
From selelnium.webdirver.support.ui import WebDriverWait  
  
From selenium.webdriver.support import expected\_conditions as EC  
  
E = WebDriverWait(driver, 5,0.5).until(  
  
EC.presence\_of\_element\_located( (By.ID, ‘kw’) )  
  
)  
  
2) 隐式等待  
  
隐式等待中的时间并非一个固定的等待时间，它并不影响脚本的执行速度。比如进行某元素的定位时，如果元素可以定位就继续执行，如果目前定位不到就以轮询的方式持续判断该元素是否被定位到，如果超过规定的时间还没定位到就抛出异常。（甲约乙去吃饭，甲在这里等着，等了10分钟乙到达，则从10分钟之后开始下一步，假如等了20分钟，则20分钟后开始下一步，如果到了最大时间30分钟还没到就抛异常）  
  
driver.implicitly\_wait(20)  
  
3) 强制等待  
  
From time import sleep  
  
Sleep(2)  
  
12. 截图  
  
driver.get\_screenshot\_as\_file('E:\sc.jpg')  
  
13. 线性测试  
  
开发维护成本高  
  
14. 模块化驱动测试  
  
将重复的操作独立为公共模块，尽量消除重复，提高测试用例的可维护性。  
  
15. 数据驱动测试  
  
数据参数化  
  
1) 读取txt文件  
  
File= open(‘E:\wtest.txt’, ‘r’)  
  
Lines = File.readlines()  
  
For line in lines:  
  
User = line.split(‘,’)[0]  
  
Passw=line.split(‘,’)[1]  
  
Print(User, Passw)  
  
2) 读取csv文件  
  
UnicodeDecodeError （未找到解决方案）  
  
3) 读取xml文件  
  
   
  
16. 关键字驱动测试  
  
脚本/数据/业务 分离  
  
   
  
17. page object设计模式  
  
是将page对象封装成一个HTML页面，通过提供的应用程序特定的API来操作页面元素，而不是在html中来搜寻对象，即提供一个易于编程的接口并隐藏窗口中底层的部件，对界面的交互细节进行封装。  
  
优点是：减少了代码的重复/提高测试用例的可读性/提高测试用例的可维护性（特别是UI频繁变化的项目）  
  
page object设置模式中，不需要在page里定位的方法中加上断言（why）  
  
   
  
18. Assert断言  
  
-断言的英文是assertion，断言检查的英文是assertion checking。  
  
-断言是指定一个程序必须已经存在的状态的一个逻辑表达式，或者一组程序变量在程序执行期间的某个点上必须满足的条件。  
  
Assert判断如果出现错误，则会中断整个测试；使用verify进行判断如果出现错误仍会继续执行直到结束。  
  
   
  
19. Dsfdsadsaf  
  
20. Dsaf倒数

# selenium+python面试题目总结



[zhusongziye](https://blog.csdn.net/zhusongziye) 2018-11-10 22:11:41  2000  收藏 20

分类专栏： [软件测试面试宝典](https://blog.csdn.net/zhusongziye/category_7131189.html) 文章标签： [selenium python面试题目总结](https://so.csdn.net/so/search/s.do?q=selenium%20python%E9%9D%A2%E8%AF%95%E9%A2%98%E7%9B%AE%E6%80%BB%E7%BB%93&t=blog&o=vip&s=&l=&f=&viparticle=)

**✎✎✎第一小节>>**

**1. WebDriver原理**

webDriver是按照client/server模式设计，client就是我们的测试代码，发送请求，server就是打开的浏览器来打开client发出的请求并做出响应。

具体的工作流程：

·webdriver打开浏览器并绑定到指定端口。启动的浏览器作为remote server.

·client通过commandExecuter发送httpRequest给remote server的的侦听端口（the webdriver wire control通信协议）

·Remote server根据原生的浏览器组件来转化为浏览器的native调用。

备注：启动浏览器的时候，使用的是http: hyper test transfer protocol

**2. Selenium 特点和组成**

Selenium是一个针对web应用的开源测试框架，它的测试用例可以用html table或者html 代码或者编程语言进行开发，而且他能在几乎所有的现在的浏览器上执行。Selenium可以被部署到windows.linux.mac平台上，支持的语言有java/python/ruby/.net/perl

特点：

·支持录制回放

·多种方式对页面元素进行定位并操作

·能够及进行debug和设置断点

·能够把测试脚本保存成html/ruby或者其他语言

·支持user-extensions.js形式的用户扩展

·能够进行页面的断言

1) Selenium IDE: 是firefox的插件，可以录制/回放/测试脚本

2) Selenium RC：支持程序语言（java/C#/python/ruby）编写测试用例; selenium和其他测试框架集成，比如java环境下将selenium和Junit集成，利用selenium来书写测试用例，用Junit来实现测试用例的自动化运行。

3) Selenium Core：是selenium的核心，是有js和html文件组成，它是selenium IDE和selenium RC的核心引擎。

4) Selenium Grid：是selenium的扩展，可以将一批selenium脚本分配到不同的测试机上同步运行。即selenium grid可以在不同的主机上建立主节点(hub)和分支节点(node), 可以使主节点上的测试用例在不同的分支节点上运行。对于不同的节点来说，可以搭建不同的测试环境（操作系统/浏览器）。

**3.如何提高自动化脚本稳定性**

找原因

1)网速原因，增加时间等待

2)函数原因，尽量少使用容易冲突的函数

3)配置testNG实现多线程，在编写测试用例的时候，一定要实现松耦合，在服务器允许的情况下尽量设置多线程运行，提高执行速度。

**4.高质量自动化脚本特点**

业务和代码分离，封装型好。

自动化用例耦合性低，独立性强，易于扩展维护

**5.自动化测试缺陷**

不稳定

可靠性不强

不易维护

成本与收益

**6.自动化用例的执行策略**

根据自动化执行的目的来决定

（1）自动化测试用例执行用来监控，可以设置用例定时执行；

（2）用于回归测试，可以把测试用例设置成触发式执行；

（3）不需要经常执行的测试用例，可以由人工执行。

**7. 提高selenium脚本的执行速度方式**

（1）减少操作步骤，减少不必要的操作；

（2）中断页面加载，如果页面加载内容过多并且加载的内容不影响我们测试，可以设置超时时间，中断页面加载；

（3）设置等待时间的时候，可以sleep()固定的时间，检测到元素出现后中断等待也可以提高速度；

（4） 配置testNG实现多线程，在编写测试用例的时候，一定要实现松耦合，在服务器允许的情况下尽量设置多线程运行，提高执行速度。

**8.元素定位**

1) Id定位

find\_element\_by\_id(‘kw’)   //通过id属性定位

2) Name定位

find\_element\_by\_name(‘wd’)    //通过名字进行定位，名字一般不唯一，一般不能通过其定位

3) ClassName定位

find\_element\_by\_class\_name(‘cn’)   //通过类名进行定位

4) TagName定位

find\_element\_by\_tag\_name(‘input’)  //标签，一般用于一类元素的定位

5) LinkText定位

find\_element\_by\_link\_text(‘新闻’)  //文本链接定位

6) PartialLinkText定位

find\_element\_by\_partial\_link\_text(‘一个很长的’) //文本链接的部分文字

7) 绝对路径定位find\_element\_by\_xpath(‘/html/body/div/div[2]/form/span/input[2]’) //绝对地址定位，利用层级定位

find\_element\_by\_xpath(‘//input[@name=’n’ and id=’kw’]’) //当前页面某个目录下 name为n，id为kw的input标签，利用元素属性进行定位

find\_element\_by\_xpath(‘//\*[@id=kw]’) //其中\*表示任何属性，表示id为kw的任何属性的元素

find\_element\_by\_xpath(‘//span[@class=’c’]/input’) //层级和属性结合进行定位

find\_element\_by\_xpath(‘//a[contains(href , ’test’)]’)

8) 相对路径定位

相对路径定位一般比绝对路径定位快。

find\_element\_by\_css\_selector(‘.class1’) // 类名定位 .class

find\_element\_by\_css\_selector(‘#id’) //id定位 #id

find\_element\_by\_css\_selector(‘input’) // 标签名定位

find\_element\_by\_css\_selector(‘[autocomplete=off]’) //属性定位

find\_element\_by\_css\_selector(‘span>input’) //标签名为span的input子元素

**✎✎✎第二小节**

9) 定位一组元素

Checkboxes = find\_elements\_by\_tag\_name(‘checkbox’)

for checkbox in Checkboxes:

checkbox.click()  # 通过for循环全选所有的checkbox

Checkboxes.pop().click()  #取消勾选一些checkbox，pop(-1)=pop()表示一组元素的最后一个；pop(0)      表示一组元素的第一个，pop(1)表示一组元素的第二个，以此类推

10) 父子/兄弟/相邻节点的定位

·父->子

Find\_element\_by\_id(‘parent’).find\_element\_by\_tag\_name(‘input’) 串联寻找

Find\_element\_by\_xpath(‘//input/div[@name=’n’]/span/input[1]’)  xpath的层级

Find\_element\_by\_css\_selector(‘div#b>input’) css selector的父子关系·子->父

Find\_element\_by\_xpath(‘//div[@id=’c’]/../..’) 其中//div[@id=’c’]的父亲的父亲

·哥哥->弟弟

Find\_element\_by\_xpath(‘//div[@id=’c’’/../div[1]) 找到弟弟后找父亲然后找哥哥

Find\_element\_by\_css\_selector(‘div#d+div’) 表示id=d的div后面紧跟着的div

Find\_element\_by\_css\_selector(‘div#d~div’) 表示id=d的div后面同一级所有div

11) 二次定位

Element1 = driver.find\_element\_by\_id(‘ID1’)

Element2 = Element1.find\_element\_by\_id(‘ID2’)

12) display: none 和hidden的区别

共同点：把网页中的某个元素隐藏起来；他们在selenium中都是定位不到的。

区别：display:none不为隐藏的对象保留其物理空间，该对象在这个页面上彻底

失，看不到/摸不到；hidden使对象在网页上不可见，但该对象在网页中依然占有空间，看不到/摸得到。

备注：isElementPresent()来判断元素是否存在。

**9.常见控件使用**

1) link/button:

element.click()

注：ajax不能使用clickAndWait()函数，因为cickAndWait函数会在click之后等待页面重新加载完成，而ajax是部分刷新，所以这个页面不能重新加载完成。

2) Textbox:

element.send\_keys(‘test’)

3) Upload

Driver.find\_element\_by\_id(‘id’).send\_keys(‘D\test.txt’)

4) Mouse Event:

e1=driver.find\_element\_by\_id(‘kw’)

ActionChains(driver).double\_click()/.context\_click()/.drag\_and\_drop()/.move\_to\_eleme        nt()/.perform()   # 双击/右击/拖动/悬停/执行actionChains中的行为

 5) Dropdown:

·<Select>标签的下拉菜单

from selenium.webdriver.support.ui import Select

Select(driver.find\_element\_by\_id(‘gender’)).select\_by\_index(1)

Select(driver.find\_element\_by\_id(‘gender’)).select\_by\_value(“2”)

Select(driver.find\_element\_by\_id(‘gender’)).select\_by\_visible\_text(“Male”)

·非<Select>标签的下拉菜单

Dropdown1 = driver.find\_element\_by\_id(‘id’) #先定位到dropdown

Dropdown1.find\_element\_by\_id(“li2\_input\_2”) #再定位到dropdown中的值

 6) Alert:

driver.switch\_to.alert.accept()//接受

driver.switch\_to.alert.dismiss() //取消

Message=driver.switch\_to.alert.text  //获取弹窗的文本消息

driver.switch\_to.alert.send\_keys(‘hello’) //输入值,这个在alert和confirm中不可用

 7) Window

Driver.refresh() 刷新

Driver.back() 后退

Driver.forward() 前进

Driver.maximize\_window() 最大化

Driver.set\_window\_size(100,200) 设置窗口大小

Driver.switch\_to.window(searchwindow)

 8) Frame

Driver.switch\_to.frame(ReferenceFrame)

Driver.switch\_to.parent\_frame()  # frame需要一级一级切

Driver.switch\_to.default\_content()

**✎✎✎第三小节**

**10.等待**

1) 显式等待

等待某个条件成立时继续执行，每隔一段时间进行检测，超出最大时间    则抛出异常（程序每隔X秒看一眼，如果条件成立，执行下一步，否则等待，超出最大    时间，抛出异常）

From selenium.webdriver.common.by import By

From selelnium.webdirver.support.ui import WebDriverWait

From selenium.webdriver.support import expected\_conditions as EC

E = WebDriverWait(driver, 5,0.5).until(EC.presence\_of\_element\_located( (By.ID, ‘kw’) ))

2) 隐式等待

隐式等待中的时间并非一个固定的等待时间，它并不影响脚本的执行速度。比如进行某元素的定位时，如果元素可以定位就继续执行，如果目前定位不到就以轮询的方式持续判断该元素是否被定位到，如果超过规定的时间还没定位到就抛出异常。（甲约乙去吃饭，甲在这里等着，等了10分钟乙到达，则从10分钟之后开始下一步，假如等了20分钟，则20分钟后开始下一步，如果到了最大时间30分钟还没到就抛异常）

driver.implicitly\_wait(20)

3) 强制等待

From time import sleep

Sleep(2)

**11.截图**

driver.get\_screenshot\_as\_file('E:\sc.jpg')

**12.线性测试**

开发维护成本高

**13.模块化驱动测试**

将重复的操作独立为公共模块，尽量消除重复，提高测试用例的可维护性。

**14.数据驱动测试**

数据参数化

1) 读取txt文件

File= open(‘E:\wtest.txt’, ‘r’)

Lines = File.readlines()

For line in lines:

User = line.split(‘,’)[0]

Passw=line.split(‘,’)[1]

Print(User, Passw)

2) 读取csv文件

UnicodeDecodeError （未找到解决方案）

3) 读取xml文件

**15.关键字驱动测试**

脚本/数据/业务 分离

**16. page object设计模式**

是将page对象封装成一个HTML页面，通过提供的应用程序特定的API来操作页面元素，而不是在html中来搜寻对象，即提供一个易于编程的接口并隐藏窗口中底层的部件，对界面的交互细节进行封装。

优点是：减少了代码的重复/提高测试用例的可读性/提高测试用例的可维护性（特别是UI频繁变化的项目）

page object设置模式中，不需要在page里定位的方法中加上断言（why）

**17.Assert断言**

-断言的英文是assertion，断言检查的英文是assertion checking。

-断言是指定一个程序必须已经存在的状态的一个逻辑表达式，或者一组程序变量在程序执行期间的某个点上必须满足的条件。

Assert判断如果出现错误，则会中断整个测试；使用verify进行判断如果出现错误仍会继续执行直到结束。

python 接口自动化测试-----常见面试题汇总

码农技术客栈 2020-09-21 10:47:59 3390 收藏 95

分类专栏： 自动化测试 文章标签： python 软件测试

版权

1、软件接口是什么？

程序不同模块之间传输数据并作处理的类或函数

2、HTTP 和 HTTPS 协议区别？

答： https 协议需要到 CA（Certificate Authority，证书颁发机构）申请证书，一般免费证书 较少，因而需要一定费用； http 是超文本传输协议，信息是明文传输，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； 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；

3.HTTPS 在哪一层？

以前我面试很喜欢提网络协议的问题，有朋友说我装 X，不实用。稍有点研究网络知识，实 际就不难回答

答：HTTPS 在应用层

4、get 和 post 区别是什么？

答：POST 和 GET 都是向服务器提交数据，并且都会从服务器获取数据。 区别： 1）传送方式：get 通过地址栏传输，post 通过报文传输 2）传送长度：get 参数有长度限制（受限于 url 长度），而 post 无限制 3）GET 产生一个 TCP 数据包（对于 GET 方式的请求，浏览器会把 http header 和 data 一并 发送出去，服务器响应 200 返回数据），POST 产生两个 TCP 数据包（对于 POST，浏览器先 发送 header，服务器响应 100 continue，浏览器再发送 data，服务器响应 200 ok 返回数 据） 4）get 请求参数会被完整保留在浏览历史记录里，而 post 中的参数不会被保留 5）在做数据查询时，建议用 GET 方式；而在做数据添加、修改或删除时，建议用 post 方式

5、常见的 POST 提交数据方式

答： 主要有四种方式：application/x-www-form-urlencoded、multipart/form-data、 application/json、text/xml 等。

6、什么是 Http 协议无状态协议?怎么解决 HTTP 协议无状态协议

答： 无状态是指协议对于事务处理没有记忆能力，服务器不知道客户端是什么状态。即我们给服 务器发送 HTTP 请求之后，服务器根据请求，会给我们发送数据过来，但是，发送完，不会 记录任何信息。HTTP 是一个无状态协议，这意味着每个请求都是独立的，Keep-Alive 没能 改变这个结果。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导 致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。HTTP 协议这种特性有优点也有缺点，优点在于解放了服务器，每一次请求“点到为止”不会造成 不必要连接占用，缺点在于每次请求会传输大量重复的内容信息。客户端与服务器进行动态 交互的 Web 应用程序出现之后，HTTP 无状态的特性严重阻碍了这些应用程序的实现，毕竟 交互是需要承前启后的，简单的购物车程序也要知道用户到底在之前选择了什么商品。于是， 两种用于保持 HTTP 连接状态的技术就应运而生了，一个是 Cookie，而另一个则是 Session。

7、cookie 和 session 的区别

答： cookie 数据存放在客户的浏览器上，session 数据放在服务器上 cookie 不是很安全，别人可以分析存放在本地的 cookie 并进行 cookie 欺骗，考虑到安全 应当使用 session session 会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能，考虑 到减轻服务器性能方面应当使用 cookie 单个 cookie 保存的数据不能超过 4K，很多浏览器都限制一个站点最多保存 20 个 cookie 可以将登陆信息等重要信息存放为 session；其他信息需要保存，可以放在 cookie

8、请求接口中常见的返回状态码

答：

1xx – 信息提示（表示临时的响应。客户端在收到常规响应之前，准备接收一个或多个 1xx 响应）

2xx – 成功（表明服务器成功地接受了客户端请求）

3xx – 重定向（客户端浏览器必须采取更多操作来实现请求。例如，浏览器可能不得不请 求服务器上的不同的页面，或通过代理服务器重复该请求）

4xx – 客户端错误（发送错误，客户端有问题。例如，客户端请求不存在的页面，客户端 未提供有效的身份证验证信息） 5xx – 服务器错误（服务器由于遇到错误而不能完成该请求）

常见的返回码有：

 200 OK - [GET]：服务器成功返回用户请求的数据

 201 CREATED - [POST/PUT/PATCH]：用户新建或修改数据成功 

202 Aceepted - []：表示一个请求已经进入后台排队（异步任务）

 204 NO CONTENT - [DELETE]：用户删除数据成功

 400 INVALID REQUEST - [POST/PUT/PATCH]：用户发出的请求有错误，服务器没有进行 新建或修改数据的操作  401 Unauthorized -[] ：表示用户没有权限（令牌、用户名、密码错误）

 403 Forbidden -[] ：表示用户得到授权（与 401 错误相对），但是访问被禁止

 404 NOT FOUND -[]：用户发出的请求针对得到是不存在的记录，服务器没有进行操作， 该操作是幂等的

 406 Not Acceptable - [GET]：用户请求的格式不可得（比如用户请求 JSON 格式，但 是只有 XML 格式）

 500 INTERNAL SERVER ERROR - [\*]：服务器发生错误，用户将无法判断发出的请求是 否成功

9、什么是 DNS？

答：DNS 是域名系统 (Domain Name System)，DNS 是用来做域名解析的,它会在你上网输入 网址后，把它转换成 IP，然后去访问对方服务器；没有它，你想上百度就要记住百度的 IP， 但有了 DNS 的处理，你只需要记住对应网站的域名，即网址就可以了。

10、请问你们公司是如何做接口测试的？

答： 接口测试实际跟一般测试不同就是测试用例的设计部分。 ①获取接口规范。

②设计接口测试功能用例（主要从用户角度出发看接口能否实现业务需求，用例设计就是黑 盒用例那一套）。

③各种入参验证（正常情况，异常情况包括输入参数个数不对，类型不对，可选/必选，还 有考虑参数有互斥或关联的情况）。

④接口返回值各种验证（符合接口文档需求）

⑤了解接口实现逻辑，实现逻辑覆盖（语句/条件/分支/判定/…）

⑥接口能并发执行吗、安全吗，性能满足要求吗?

⑦采用工具或者自写代码来验证。

⑧发现问题跟功能测试一样，该报 bug 报 bug，该跟踪状态的跟踪状态。

11、怎么设计接口测试用例？

答： 通常，设计接口测试用例需要考虑以下几个方面：

①是否满足前提条件 有些接口需要满足前提，才可成功获取数据。常见的，需要登录 Token 逆向用例：针对是否满足前置条件（假设为 n 个条件），设计 0~n 条用例

②是否携带默认值参数 正向用例：带默认值的参数都不填写、不传参，必填参数都填写正确且存在的“常规”值， 其他不填写，设计 1 条用例

③业务规则、功能需求 这里根据时间情况，结合接口参数说明，可能需要设计 N 条正向用例和逆向用例

④参数是否必填 逆向用例：针对每个必填参数，都设计 1 条参数值为空的逆向用例

⑤参数之间是否存在关联 有些参数彼此之间存在相互制约的关系

⑥参数数据类型限制 逆向用例：针对每个参数都设计 1 条参数值类型不符的逆向用例

⑦参数数据类型自身的数据范围值限制 正向用例：针对所有参数，设计 1 条每个参数的参数值在数据范围内为最大值的正向用例

12、你做接口测试，测什么？

答：

可用性测试 根据约定的协议、方法、格式内容，传输数据到接口经处理后返回期望的结果：

 接口功能是否正确实现；

 返回值测试 - 返回值除了内容要正确，类型也要正确，保证调用方能够正确地解析；

 参数值边界值、等价类测试； 错误和异常处理测试

 输入异常值（空值、特殊字符、超过约定长度等），接口能正确处理，且按预期响应；

 输入错误的参数，接口能正确处理，并按预期响应；

 多输入、少输入参数，接口能正确处理，且按预期响应

 错误传输数据格式（如 json 格式写成 form 格式）测试； 安全性测试，主要指传输数据的安全性：

 敏感数据（如密码、秘钥）等是否加密传输；

 返回数据是否含有敏感数据，如用户密码、完整的用户银行账号信息等；

 接口是否对传入的数据做安全校验，如身份 ID 加 token 类似校验；

 接口是否防止恶意请求（如大量伪造请求接口致使服务器崩溃）； 性能测试，如接口的响应时间、并发处理能力、压测处理情况：

 并发请求相同的接口（特别为 POST 请求），接口的处理情况（如插入了相同的记录导致 数据出错，引发系统故障）； 测码学院

 接口响应时长在用户可忍受的范围内；  对于请求量大的接口做压测，确定最大的瓶颈点是否满足当前业务需要；

13、平常用什么工具测接口的？

答：常用 http 协议接口测试工具，如：postman、fiddler、jmeter；webService 接口用 SoapUI、 jmeter 等。

14、没有接口文档，如果做接口测试？

本题主要考情商，通俗来说就是忽悠能力，先唬住面试官了再说，进去了也是瞎测测，随时 做好背锅的准备，当然，你肯定不能回答面试官不测(心理 mmp，脸上笑嘻嘻)，接下来就是 扯犊子时间

答：用抓包工具把接口抓取处理，然后针对性进行测试；接口中字段信息不清楚的，找时间

集中寻求开发解答。（常用抓包工具 Fiddler、Charles 等）

15、在手工接口测试或者自动化接口测试的过程中，上下游接口有数据依赖如何处理

？ 答：用一个全局变量来处理依赖的数据，比如登录后返回 token，其它接口都需要这个 token， 那就用全局变量来传 token 参数。

16、依赖于第三方数据的接口如何进行测试？

答：mock 接着面试官会问你，如果 mock 的，然后你就顺着坑继续挖，搭建 mock 服务，参考这篇 http://www.51ste.com/share/det-485.html

17、接口测试中，依赖登录状态的接口如何测试？

答：依赖登录状态的接口的本质上是在每次发送请求时需要带上 session 或者 cookie 才能 发送成功，在构建 POST 请求时添加必要的 session 或者 cookie

18、如何模拟弱网做测试

？ 答：Fiddler 和 charles 都可以模拟弱网测试，平常说的模拟丢包，也是模拟弱网测试。具

体可以看《几种弱网模拟方法，总有一种适合你》

19、你平常做接口测试的过程中发现过哪些 bug?

面试官出这个题，主要是想知道你是不是真的做过接口测试，毕竟现在很多小伙伴简历经过 包装（不包装连面试机会都没有，没办法，为了生存，能理解）

答：

常规错误，接口没实现，没按约定返回结果，边界值处理出错等。 输入异常值（空值、特殊字符、超过约定长度等），接口抛错，没做封装处理； 输入错误的参数、多输入、少输入参数，

接口可能出现的错误； 安全性问题，如明文传输、返回结果含有敏感信息，没对用户身份信息做校验，没做恶意请 求拦截等；

性能问题，如接口并发插入多条相同操作，响应时间过长，接口压测出现瓶颈等； 测码学院

20、当一个接口出现异常时候，你是如何分析异常的？

答： 先抓包，用 fiddler（charles）工具抓包，或者浏览器上 F12 调试工具；APP 上的话，那就 用 Fiddler 做代理，通过手机设置代理去看请求和返回报文； 查看后端日志，如 Linux 系统通过 xhell 连上服务器，查看接口日志，查看是否有报错信息 （命令：tail -f 日志文件）；

21、如何分析一个 bug 是前端还是后端的？

答： 平常提 bug 的时候，前端开发和后端开发总是扯皮，不承认是对方的 bug。 这种情况很容易判断，先抓包看请求报文，对着接口文档，看请求报文有没问题，有问题就 是前端发的数据不对； 请求报文没问题，那就看返回报文，返回的数据不对，那就是后端开发的问题咯。

22、你们做接口测试自动化吗？

答：现在针对大量应用，普遍推崇做接口测试自动化，维护成本低、收益高。常用的工具有 许多，如 Jmeter、Robot Framework、pytest 等。

23、列出几个 JMeter 监听器？

一些 JMeter 监听器是： 集合报告 汇总报告 查看结果树 用表格查看结果 图形结果 BeanShell Listener 摘要报告等

24、在 python 中进行数据驱动测试

在 unittest 中，没有自带的数据驱动，我们得借助 ddt 来实现，首先，我们得在 python 运行环境中安装 ddt，用下列命令安装 pip install ddt 另 外 一 个 测 试 框 架 pytest ， 它 自 带 数 据 驱 动 实 现 ， 是 通 过 @pytest.mark.parametrize(argnames,argvalues) 来实现参数化的。 也可以根据自己需求用 python 实现数据的读取和驱动。

25、接口自动化中的关联怎么处理？

把上一个请求返回的结果传入到下一个请求的参数中，将请求的结果反射到一个类属性（使 用 setattr（）函数），下一个请求去调用这个类属性

26、自动化测试怎么校验结果？

断言 ，预期结果与实际结果对比

数据库校验，根据测试场景来查询数据库里的数据和请求之前的数据进行比对 测码学院

27、自动化使用的测试框架是什么？

简述自动化框架的设计、维护

测试框架：python+unittest+requests+ddt+openpyxl+pymysql+logging

python：入门简单，语法简洁

unittest ：定义一个测试用例类，具体的方法来维护测试用例的生命周期，测试场景行为， 测试用例 前置场景，行为，期望结果，实际结果，断言方法，Setup teardown 方法

requests：接口调用 ，支持 http 请求的库，API 简洁，提供不同的 http 请求方法，支持 session，cookies，

ddt ：数据驱动，ddt 类装饰器，data 测试方法装饰器 unpack 解包可迭代的数据类型

普通用户，数据库，配置文件—（基础数据）

openpyxl： 数据管理 excel 管理数据，使用 openpyxl 模块来进行 excel 数据的读和写 （excle，csv, json, yaml, txt 都可以管理测试数据）

pymysql：数据库交互，数据校验

eval，json：数据格式的转换 Eval 将 python 支持的格式转换成对应的格式

logging：日志处理， 统一日志输出格式，渠道，级别，执行结果的记录，便于定位问题

jenkins：持续集成

2/框架设计思路：数据驱动+结构分层（可读性，可维护性，可扩展性）

数据驱动：将维护数据与代码分离，接口调用行为一致，针对不同的参数组合驱动不同的测 试场景，减少代码冗余

结构分层：数据层+用例层+逻辑层

数据层：测试数据的支撑 data.xls

用例层：用例的执行 test\_register.py test\_recharge.py

逻辑层：公用的方法的封装与提取 doexcle.py do\_mysql.py http\_requests.py logger.py 等模块

3/框架设计步骤：

准备测试数据： EXCEL 表准备测试用例—excel 数据的读取—参数值的替换 测码学院

发起请求：请求方法（get/post 方法进行封装—URL 的拼接（不同—参数转化为字典

拿到请求的返回值：解析返回值 code，status，msg 信息

断言

好处：

1、自动化测试用例和手工测试用例的完美结合，减少重复工作

2、配置灵活，可以自主切换测试环境，执行测试用例

3、常用功能进行封装，逻辑清晰，易于维护

4、统一执行入口，管理测试用例集：

run.py 模块通过模糊查找来选择需要执行的测试用例

5、持续集成，定时构建，快速反馈

28、具体的在这个项目中自动化怎么应用到实际的，您对自动化结果的分析

完成所有的自动化测试框架的设计和实现后，进行接口测试，然后集成到

jenkins，配置定时执行，生成 html 报表，查看测试通过率，查看接口的功能

每次发版时，进行回归测试，新功能开发未提测前

————————————————

版权声明：本文为CSDN博主「码农技术客栈」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/weixin_44602565/article/details/108704374>

接口测试面试题（一）

己兮 2021-01-22 14:52:49 202 收藏 3

分类专栏： 软件测试 接口测试 文章标签： 软件测试 接口测试 接口测试面试题

版权

时间是挤出来的，那么就冲啊，奥里给

接口测试面试题

HTTP, HTTPS协议

什么是DNS

怎么抓取HTTPS协议（Fiddler）

HTTP协议请求方式

HTTP请求报文与响应报文格式

HTTP报文

HTTP请求报文

HTTP响应报文

HTTP, HTTPS协议

HTTP

超文本传输协议，是一个基于请求与响应，无状态的，应用层的协议，常基于TCP/IP协议传输数据，互联网上应用最为广泛的一种网络协议,所有的WWW文件都必须遵守这个标准

HTTPS

HTTPS是一种通过计算机网络进行安全通信的传输协议，经由HTTP进行通信，利用SSL/TLS建立全信道，加密数据包。HTTPS使用的主要目的是提供对网站服务器的身份认证，同时保护交换数据的隐私与完整性

什么是DNS

DNS 是域名系统 (Domain Name System) 的缩写,该系统用于命名组织到域层次结构中的计算机和网络服务

怎么抓取HTTPS协议（Fiddler）

Fiddler是个很好的代理工具，可抓取协议请求用于调试

关于Fiddler抓取HTTP协议的原理和配置比较简单

第一步，Fiddler截获客户端发送给服务器的HTTPS请求，Fiddler伪装成客户端向服务器发送请求进行握手

第二步，服务器发回相应，Fiddler获取到服务器的CA证书， 用根证书公钥进行解密， 验证服务器数据签名，获取到服务器CA证书公钥。然后Fiddler伪造自己的CA证书， 冒充服务器证书传递给客户端浏览器

第三步，与普通过程中客户端的操作相同，客户端根据返回的数据进行证书校验、生成密码Pre\_master、用Fiddler伪造的证书公钥加密，并生成HTTPS通信用的对称密钥enc\_key

第四步，客户端将重要信息传递给服务器， 又被Fiddler截获。Fiddler将截获的密文用自己伪造证书的私钥解开， 获得并计算得到HTTPS通信用的对称密钥enc\_key。Fiddler将对称密钥用服务器证书公钥加密传递给服务器

第五步，与普通过程中服务器端的操作相同，服务器用私钥解开后建立信任，然后再发送加密的握手消息给客户端

第六步，Fiddler截获服务器发送的密文， 用对称密钥解开， 再用自己伪造证书的私钥加密传给客户端

第七步，客户端拿到加密信息后，用公钥解开，验证HASH。握手过程正式完成，客户端与服务器端就这样建立了"信任"

HTTP协议请求方式

标准Http协议支持六种请求方法，即：

GET：GET可以说是最常见的了，它本质就是发送一个请求来取得服务器上的某一资源。资源通过一组HTTP头和呈现数据（如HTML文本，或者图片或者视频等）返回给客户端。GET请求中，永远不会包含呈现数据。

HEAD：HEAD和GET本质是一样的，区别在于HEAD不含有呈现数据，而仅仅是HTTP头信息。

PUT：这个方法比较少见。HTML表单也不支持这个。本质上来讲，PUT和POST极为相似，都是向服务器发送数据，但它们之间有一个重要区别，PUT通常指定了资源的存放位置，而POST则没有，POST的数据存放位置由服务器自己决定。

DELETE：删除某一个资源。基本上这个也很少见，不过还是有一些地方比如amazon的S3云服务里面就用的这个方法来删除资源。

POST：向服务器提交数据。这个方法用途广泛，几乎目前所有的提交操作都是靠这个完成

OPTIONS：这个方法很有趣，但极少使用。它用于获取当前URL所支持的方法。若请求成功，则它会在HTTP头中包含一个名为“Allow”的头，值是所支持的方法，如“GET,

POST”

HTTP请求报文与响应报文格式

HTTP报文

HTTP报文是由一行一行的简单字符串组成的，HTTP报文都是纯文本，不是二进制代码

HTTP 报文包括以下三个部分：

起始行

报文的第一行就是起始行,在请求报文中用来说明要做些什么,在响应报文中说明出现了什么情况。

首部字段

起始行后面有零个或多个首部字段。每个首部字段都包含一个名字和一个值,为了便于解析,两者之间用冒号(:)来分隔。首部以一个空行结束。添加一个首部字段和添加新行一样简单。

主体

空行之后就是可选的报文主体了,其中包含了所有类型的数据。请求主体中包括 了要发送给 Web 服务器的数据;响应主体中装载了要返回给客户端的数据。起始行和首部都是文本形式且都是结构化的,而主体则不同,主体中可以包含任意的二进制数据

HTTP请求报文

请求报文(请求行/请求头/请求数据/空行)

请求行

方法字段、URL字段和HTTP协议版本

请求方法：

GET、POST、HEAD、PUT、DELETE、OPTIONS、TRACE、CONNECT

请求头(key value形式)

User-Agent：产生请求的浏览器类型。

Accept：客户端可识别的内容类型列表。

Host：主机地址

请求数据

post方法中，会把数据以key value形式发送请求，空行，发送回车符和换行符，通知服务器以下不再有请求头

HTTP响应报文

响应报文(状态行、消息报头、响应正文)

状态行

消息报头

响应正文

这篇文章到此结束，下一篇依然接口测试，谢谢

————————————————

版权声明：本文为CSDN博主「己兮」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/HONGTester/article/details/112983239>

# 接口测试常见面试题



[张哈哈爱吃肉](https://blog.csdn.net/weixin_40720301) 2020-04-17 16:10:24  5901  收藏 123

版权

一.get和post请求的区别：

get和post请求都可以向服务器发送收据

1>传送方式：get通过地址栏传送数据，post通过body传送数据

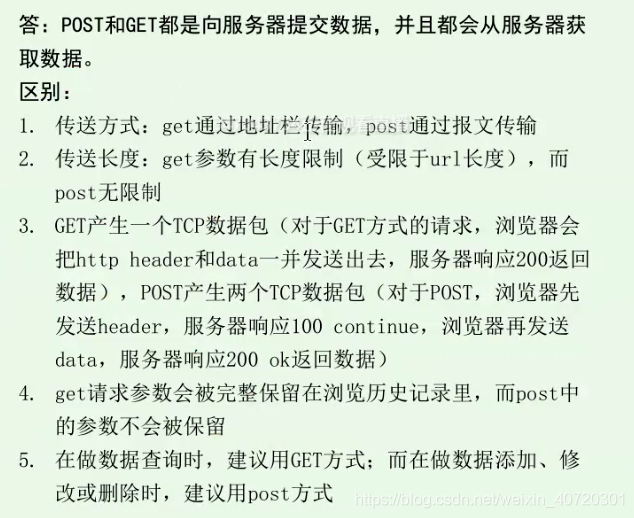
2>传送长度：get受限制，post长度不受限制

3>get 1  tcp  200     (TCP协议，数据包：TCP协议的数据包)

    post  2

get 发一次TCP包，POST发两次 一次返回100，第二次返回200

4>



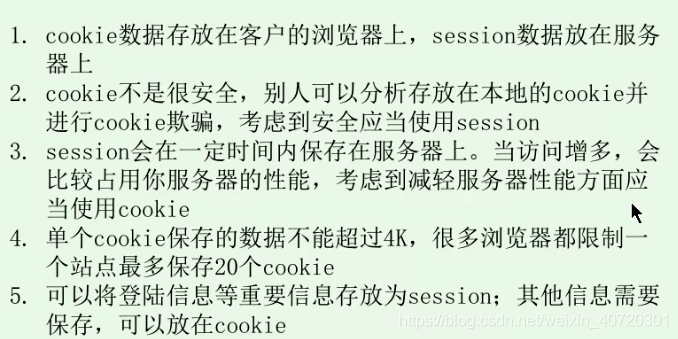
二.cookie和session的区别

    1.cookie存在本地的，session存服务器

    2.cookie不安全（可篡改）

    3.性能（session过多占用服务器空间）

    4.c<4k,20个C



三.接口常见的返回码

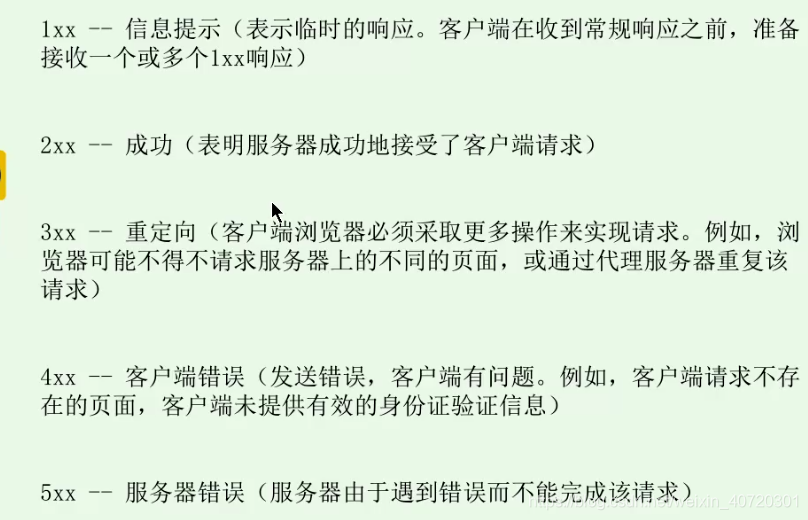
    1XX    信息

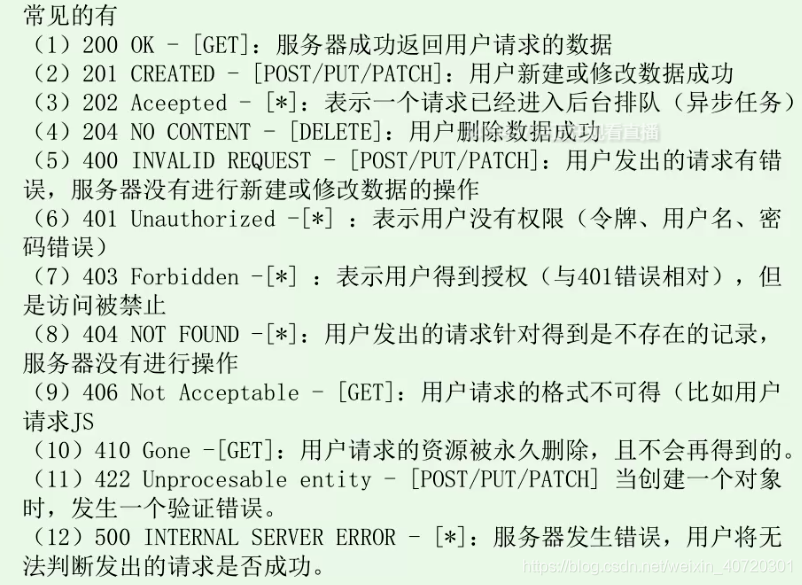
    2XX   成功    200

    3XX   重定向

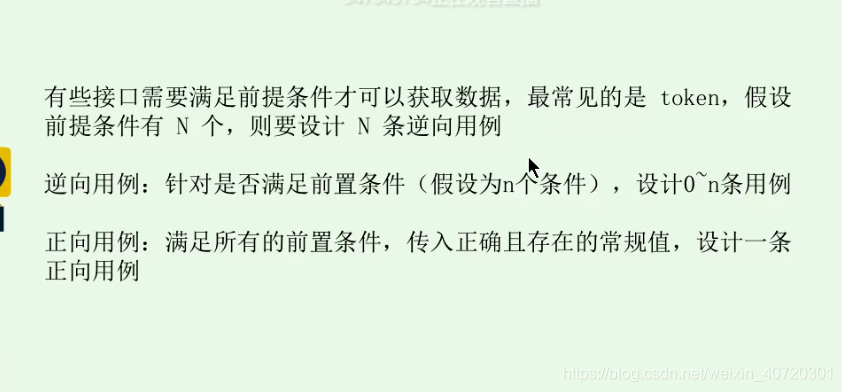
    4XX   客户端错误   400,404

    5XX   服务器错误    500





四.有前提条件的接口如何进行测试



取消订单：1.有顶单；2商家同意；3.已付款

正向：1,2,3

逆向：1！；2！；3！；

