

面试题

01. 你们公司是如何做接口测试的？

分析：

- 回答测试流程
- 按测试流程，结合实际工作描述接口测试内容
- 展现接口测试成果

答：

在我上家公司，首先针对用户提出的需求，展开 **需求分析**，由产品经理主导，出具需求文档。开发工程师根据需求文档，开始编写代码，功能测试工程师根据需求文档，设计功能测试点，撰写测试用例。接口测试人员，需要等待后端输出接口文档后才能展开工作，在等待期间，可以熟悉一下需求，准备接口测试数据。

后端开发提供接口文档后，接口测试工程师根据接口文档，**设计接口测试用例**。根据测试用例，展开 **接口测试工作**，发现BUG，使用禅道完成**提交、跟踪与管理缺陷**的工作。

等所有**测试用例执行率达到100%**，发现**普通等级以上的BUG都已经修复了**，那么我们就可以**输出接口测试的报告**，**功能测试人员**再根据我们输出的接口测试报告，进行功能测试。

在实际的接口测试工作当中，比较困难的部分是登录接口的验证码处理，我们系统中，有一个人脸识别技术，通过接口很难进行测试，这个时候，我们是通过mock的方式绕过了人脸识别的登录接口，进而完成的测试。人脸识别的验证，放在了功能阶段进行。

我们公司，使用的接口测试工具主要是Postman，使用到断言、数据关联、参数化等技术，来实现接口测试，最后使用newman来生成测试报告。也了解过 jmeter 工具和 python + requests 框架来进行接口测试。postman 工具有转换代码的功能，也可以把postman的界面操作转换为对应python代码，进行接口测试。

02. 什么时候开展接口测试？

答：

在我们公司中，我们有独立接口测试部门，等后端开发提测接口之后，就开始进行接口测试工作。

根据个人的了解，我们公司是在 **后端开发提测** 接口时，进行的接口测试。

但是有的公司，会在需求评审之后，开发负责人就会输出项目的标准接口文档，所以这些公司，会在需求评审之后，开发负责人输出标准接口文档之后，就开始了接口测试。

我们还会进行接口的持续集成，在整个接口测试通过之后，就可以进行接口的自动化测试和持续集成了，这个部分也是属于接口测试。

03. 接口测试和UI测试的工作是否重复？

答：

如果单纯只看UI测试，不看UI测试中的功能部分，那么UI测试和接口测试没有重复。

因为UI测试只关注界面之间的交互，布局，按钮的点击等等。这些UI操作都可以脱离接口来进行。

但是如果UI测试还包括功能测试，那么 UI测试 和 接口测试 在业务测试点是高度重复的。

怎么解决重复工作问题？

思路1：

- 让接口测试工程师，重点关注服务器端业务交互；
- 让UI测试工程师，重点关注界面的显示、页面之间的跳转、布局等页面元素的变化。关注主业务流程是否与服务器端的操作流程吻合。

思路2：

- 不划分功能测试和接口测试人员。所有测试工程师，都既做功能测试也做接口测试。这样可以避免资源浪费情况。

04. 接口测试框架怎么搭建的？

相当于问：你们公司的接口测试的技术框架是什么？（代码）

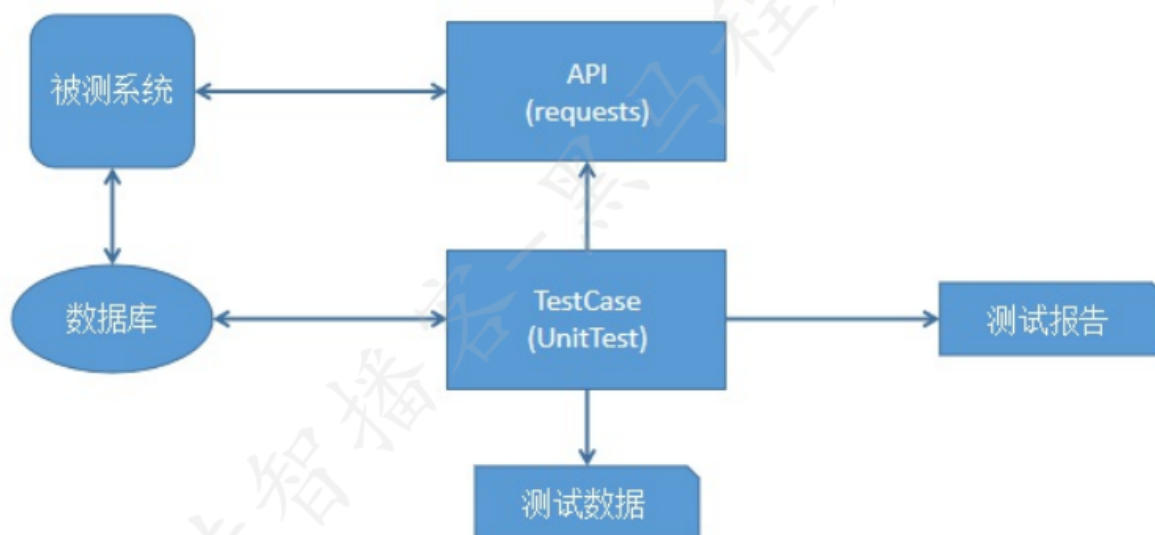
答：

我们公司使用的接口测试框架是 python + unittest + requests + parameterized 实现的

框架结构主要分为6部份。分别是：被测系统，API，测试用例，测试数据，测试报告和数据库。

他们的关系是，API用于封装被测系统的接口，测试用例使用封装的接口编写接口测试脚本，测试数据会和测试脚本分离开发，主要采用json结构来存储测试数据，使用unittest生成测试报告，

使用pymysql，进行数据库相关的操作。



05. 接口之间有依赖时怎么处理？

分析：建议，回答面试题是，尽量套用实际项目作答。如果，你不举例，面试官就会结合他们公司自己的项目任选一个有关联的接口让你分析。

答：

在我参与的上个项目中，就有依赖关系的处理，是员工模块的增、删、改、查等接口依赖“登录成功接口”，

当时，使用postman的关联技术来进行处理的。

首先是提取登录接口的令牌，然后把令牌保存到环境变量，最后在员工的增删改查4个接口中按照接口文档的要求，通过请求头的 Authorization 参数完成令牌的关联

06. 如何判断接口测试的结果（成功或失败）？

分析：可以结合实际项目回答

答：

以注册接口为例，怎么判断注册接口的测试结果是正确的呢？

有3大校验要点：

1. 判断响应数据与预期是否一致
 - 注册完成返回的响应数据，与预期数据是否一致
2. 断言数据与数据库中的数据是否一致。
 - 查询接口，数据库中的数据作为预期数据；响应数据是实际数据。
- 写入接口（增、删、改），数据库中的是实际数据，写入的数据是预期数据
 - 判断注册账号成功之后，数据库中写入用户信息，是否与预期数据一致，是否符合业务逻辑。
3. 判断业务逻辑正确性。注册成功之后，需要登录查看注册的账号是否能够登录

强调：接口测试 和 功能测试 是 相通。

如果这个面试题换成 **如何判断测试的结果**，回答的方式是一模一样的。

07. 发送HTTP请求时，传递参数的途径有哪些？

答：

URL、请求头、请求体

08. 持续集成如何做的？自动化测试多久构建一次？

答：

在之前的项目当中，我负责了人力资源管理系统登录模块和员工管理模块的持续集成工作。

我们的项目主要是使用持续集成工具，jenkins + 项目管理工具git + HTML Publisher + 邮件来进行持续集成

主要是利用源码管理，配置拉取git仓库中的接口自动化测试代码，然后使用构建命令运行项目入口文件，从而实现自动化测试。在这其中使用了构建触发器的定时构建和轮巡构建。

利用定时构建，每天05:00执行1次自动化测试。通过轮巡构建，检查自动化接口测试的代码有没有更新，有更新就触发构建。触发之后，会使用HTML Publiiser生成测试报告，并发送邮件，给测试部门所有员工，和开发负责人。

进行持续集成之后，我们可以把晚上的时间也利用起来，进行自动化的接口测试工作，提升了测试效率。

构建频率不是固定的，一般每天至少1次。如果代码变动频繁，可以会有更多次。

09. 使用工具和代码实现接口测试的区别？

答：

- 工具：
更容易上手、使用简单。能应对实际工作中大部分需求。
但，受工具本身功能限制，不适合应对对技术要求较高的场景。
- 代码：
学习难度较大，使用有一定难度，需反复练习才能掌握。
但，灵活度非常高。适合任何场景。

在实际工作当中，日常使用以工具为主。代码是高薪的基础，是软件测试行业未来发展的方向。

10. cookies和session的区别

答：

- 位置不同： a) cookie数据存放在客户的浏览器上， b) session数据放在服务器上。
- 安全性： a) 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；
b) session安全性更高。
- 占用空间： a) session会在一定时间内保存在服务器上。当访问增多，会比较占用服务器的性能 b) 考虑到减轻服务器性能方面，应当使用COOKIE。
- 大小不同 a) 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。 b) session是无限量的

11. 什么是HTTP协议，它和HTTPS有什么区别？

答：

- HTTP协议 是超文本传输协议，被用于在Web浏览器和网站服务器之间传递信息。HTTP协议以明文方式发送内容，不提供任何方式的数据加密，如果攻击者截取了Web浏览器和网站服务器之间的传输报文，就可以直接读懂其中的信息，因此HTTP协议不适合传输一些敏感信息，比如信用卡号、密码等。
- HTTPS（安全套接字层）传输协议,为了数据传输的安全，HTTPS在HTTP的基础上加入了SSL协议，SSL依靠证书来验证服务器的身份，并为浏览器和服务器之间的通信加密。

区别：

1. HTTP 的URL 以http:// 开头，而HTTPS 的URL 以https:// 开头
2. HTTP 是不安全的明文传输，而 HTTPS 是安全的加密传输
3. HTTP 标准端口是80，而 HTTPS 的标准端口是443
4. 在OSI 网络模型中，HTTP工作于应用层，而HTTPS 工作在传输层
5. HTTP 无法验证身份，而依赖于SSL证书的HTTPS 可验证服务器身份
6. HTTP无需证书，而HTTPS 需要正规CA wosign 颁发的SSL证书
7. HTTP协议：免费申请,而HTTPS协议：需要到ca申请证书，一般免费证书很少，需要交费

12. 场景分析

问题：在发布文章模块中，输入正确的标题、内容、频道等信息后，点击“发表”按钮后没有任何反应。请描述如何分析定位存在的问题？



答：

使用fiddler抓包工具，抓取点击“发表”按钮后产生的数据报文。

1. 点击“发表”按钮后，没有产生数据报文。----- 前端问题
2. 点击“发表”按钮后，产生报文，但数据有误。----- 前端问题
3. 点击“发表”按钮后，产生报文，服务器返回数据有误，导致前端不能正确响应。----- 前端、后端都有问题（服务器返回数据有误，前端应该给出提示，后端不应该返回错误的数据）
4. 点击“发表”按钮后，产生报文，服务器返回数据正确，前端没有反应。----- 前端问题
5. 点击“发表”按钮后，产生报文，但是服务器无响应，前端没有反应。----- 前端问题，也有可能是后端。（要进入后台查看这个时间节点是否产生了请求日志，如果有日志，但是服务器没有反应，那么服务器问题，如果没有日志，说明请求根本没有发给服务器，属于网络问题，或者URL错）
6. 其他问题（页面卡死，等等）

13. 简单描述你测试过的项目

答：

在上家公司，我有幸参与到了IHRM人力资源管理系统测试工作中，是公司购买二次开发的系统。里面集成了XX、XX、XX等功能模块。能够简化流程，快速响应和协作，提升办公效率。

在IHRM项目中，我主要负责登录、员工、部门、权限等模块的接口+功能的测试工作，一共测有近30个接口。共设计了约一千个测试用例，测试发现BUG二百多个，其中严重等级的BUG十几个，普通等级BUG一百多个，易用性相关BUG一百多个。在一定程度上推动了后端接口的质量的提升的。

最终，项目0BUG上线，全公司300多人连续使用1年没有出现严重的问题。保证了产品的质量。

14. Get和Post的区别

答：

1. 在安全性上，GET明文显示参数，参数直接暴露在URL上，POST是加密存储，安全性高。
2. 在数据大小上，GET请求的参数数据长度是有限制的，而POST请求的数据长度没有限制
3. 在编码上，GET请求只支持utf编码，而POST就支持多种编码方式
4. GET请求的参数在HTTP中是通过url传递，POST请求的数据是通过请求体传递
5. 支持的字符类型不一样，GET只支持ASCII码数据，而POST支持多种数据类型。所以中文在GET请求当中，会被编码。
6. 传输速度不一样，GET传输速度快。
7. GET请求无法传递二进制数据到服务器，而POST可以

15. http报文由哪些部分组成

答：

HTTP报文有请求报文和响应报文两种：

- HTTP请求报文由请求行（request line）、请求头（header）、空行和请求数据4个部分组成。
- HTTP响应报文由响应行，响应头，空行和响应体4部分组成。

16. 你们公司数据库的端口号是多少？

答：

首先，说明在公司的项目中用到的是那种数据库，再把答案往里带。

关系型数据库

1. **MySQL数据库，默认端口是: 3306;**
2. Oracle数据库，默认端口号为: 1521;
3. Sql Server数据库，默认端口号为: 1433;
4. DB2数据库，默认端口号为: 5000;
5. PostgreSQL数据库，默认端口号为: 5432;

NoSql 数据库（非关系型）：

1. MongoDB数据库，默认端口号: 27017;
2. **Redis数据库，默认端口号: 6379;**
3. Memcached 数据库，默认端口号: 11211;

17. 如果开发认为你提交的BUG不是BUG，你会怎么处理？

答：

1. 首先明确开发说不是bug的理由，告知开发bug的判断依据，协商解决（坚决不要起冲突）。
2. 找产品经理确认是否有需求变更
3. 如果存在测试环境问题，在特定环境下复现bug，并在测试报告中明确指出bug复现场景。
4. 明确所提交的bug级别，是否影响主要功能。
5. 如果使用场景复杂，但我们坚信确为bug，可以知会到测试经理，让测试经理判定。

18. 测试全新项目，怎么开展工作？

领导让你去参与一个全新项目的测试，只给了你项目名、这个时候你应该怎么开展工作？（例如：王思炊，这里有一个生死簿管理系统，链接：<http://www.bknks.com>，我现在有事要忙，你去测一下）

答：

1. 找产品、开发拿到需求文档和对应接口文档。对需求进行分析，要求与开发的理解达成一致。
2. 重点，思考下测试思路。应该从哪方面下手，有哪些必须覆盖到的测试点，支持哪些平台，有哪些不同的常用场景，是否需要考虑到稳定性、性能等等，针对待测项目进行整体的测试规划。
3. 根据自己思路分析结果，解析接口文档。
4. 按接口文档的解析结果，结合需求设计接口测试用例。
5. 执行测试用例。
6. 缺陷的管理与跟踪
7. 输出测试报告
8. 按需看是否需要实现接口的自动化和持续集成

19. 已知一个接口，设计测试点

该接口的URL、请求方法、请求头都不用关心。它有一个参数，名为text。参数值需要输入一个字符串，用空格区分每个单词。当输入字符串并提交后，服务器会自动对字符串中的单词进行排序，并显示在响应数据中，规则如下：

- 最长的排第1个，最短的排最后一个
- 如果有相同长度的单词，按照ASCII码的大小依次比较每个相同位置的单词，大的排前面，小的排后面。参数是请求体：{"text":"字符串"}，请针对该接口设计接口测试用例

答：

1. 测试只有1个单词 {"text":"helloworld "}, 预期：只显示1个单词
2. 测试两个单词长度不同时的场景，预期：长的排第1个，短的排第2个
3. 测试两个单词长度、内容相同场景，预期：正常显示，不分先后
4. 测试两个单词长度相同，最后一个字母不同场景，预期：ASCII码大的在前，ASCII码小的在后
5. 测试3个单词长度不同，预期：长的排第1个，短的排第3个，中间的排第2个
6. 测试3个单词长度相同、内容相同场景，预期：正常显示，不分先后
7. 测试3个单词长度相同，最后一个字母都不同场景，预期：ASCII码大的在前，ASCII码小的在后
8. 测试text能输入的最大长度的单词：输入65535个字母的‘不存在单词’ 预期：能正常显示
9. 测试无网络情况下发送请求
10. 测试发送请求后，服务器响应超时场景