



接口测试面试题汇总（46）

目录

一、 HTTPS 和 HTTP 的区别	4
二、 HTTPS 的工作原理	4
三、 客户端在使用 HTTPS 方式与 Web 服务器通信时有以下几个步骤：	5
四、 HTTPS 的优点	5
五、 HTTPS 的缺点	6
六、 http 切换到 HTTPS	7
七、 提到 UI 级别测试和 API 测试之间的关键区别？	7
八、 接口测试用例的编写要点有哪些？	8
九、 接口自动化测试的流程？	9
十、 POST 和 GET 有什么区别？	9
十一、 常见的 HTTP Header 及其作用	10
十二、 Session 与 Cookie 有什么区别？	13
十三、 HTTP 通讯过程中，是客户端还是服务端主动断开连接？	13
十四、 HTTP 有哪些请求方法？	14
十五、 Cookie 保存在哪里？	15
十六、 简述 TCP/IP 的三次握手和四次挥手	15
十七、 TCP 和 UDP 有什么区别	19
十八、 什么是 TCP/IP？	20



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

十九、json 和字典 dict 的区别？	21
二十、测试的数据你放在哪？	23
二十一、什么是数据驱动，如何参数化？	25
二十二、下个接口请求参数依赖上个接口的返回数据	25
二十三、依赖于登录的接口如何处理	26
二十四、依赖第三方的接口如何处理	26
二十五、不可逆的操作，如何处理，比如删除一个订单这种接口如何测试	26
二十六、接口产生的垃圾数据如何清理	27
二十七、一个订单的几种状态如何全部测到，如：未处理，处理中，处理失败，处理成功	27
二十八、python 如何连接数据库操作？	28
二十九、在 API 测试中测试的常用协议是什么？	28
三十、接口测试的步骤有哪些？	29
三十一、接口测试中依赖登录状态的接口如何测试？	29
三十二、依赖于第三方数据的接口如何进行测试？	29
三十三、解释什么是 SOAP？	29
三十四、解释什么是 REST API？	30
三十五、API Builder 如何工作？	30
三十六、解释什么是 TestApi？	30
三十七、什么是使用 runscope 的 API 测试？	31
三十八、解释 API 测试设计的原理是什么？	31
三十九、API 测试发现的 Bug 类型是什么？	32



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

四十、TCP 长连接和短连接的区别	33
四十一、Session 和 cookie 的区别	36
四十二、为什么要有 cookie，它的作用是什么？	37
四十三、Cookie 测试的测试点	38
四十四、cookie 的缺点	40
四十五、session 的作用	41
四十六、cookie 与 session 的区别：	43

一、HTTPS 和 HTTP 的区别

主要如下：

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

二、HTTPS 的工作原理

我们都知道 HTTPS 能够加密信息，以免敏感信息被第三方获取，所以很多银行网站或电子邮箱等等安全级别较高的服务都会采用 HTTPS 协议。

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

三、客户端在使用 HTTPS 方式与 Web 服务器通信时有以下几个步骤：

(1) 客户使用 https 的 URL 访问 Web 服务器，要求与 Web 服务器建立 SSL 连接。

(2) Web 服务器收到客户端请求后，会将网站的证书信息（证书中包含公钥）传送一份给客户端。

(3) 客户端的浏览器与 Web 服务器开始协商 SSL 连接的安全等级，也就是信息加密的等级。

(4) 客户端的浏览器根据双方同意的安全等级，建立会话密钥，然后利用网站的公钥将会话密钥加密，并传送给网站。

(5) Web 服务器利用自己的私钥解密出会话密钥。

(6) Web 服务器利用会话密钥加密与客户端之间的通信。

四、HTTPS 的优点

尽管 HTTPS 并非绝对安全，掌握根证书的机构、掌握加密算法的组织同样可以进行中间人形式的攻击，但 HTTPS 仍是现行架构下最安全的解决方案，主要有以下几个好处：



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

(1) 使用 HTTPS 协议可认证用户和服务器，确保数据发送到正确的客户机和服务器；

(2) 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，要比 http 协议安全，可防止数据在传输过程中不被窃取、改变，确保数据的完整性。

(3) HTTPS 是现行架构下最安全的解决方案，虽然不是绝对安全，但它大幅增加了中间人攻击的成本。

(4) 谷歌曾在 2014 年 8 月份调整搜索引擎算法，并称“比起同等 HTTP 网站，采用 HTTPS 加密的网站在搜索结果中的排名将会更高”。

五、HTTPS 的缺点

虽然说 HTTPS 有很大的优势，但其相对来说，还是存在不足之处的：

(1) HTTPS 协议握手阶段比较费时，会使页面的加载时间延长近 50%，增加 10%到 20%的耗电；

(2) HTTPS 连接缓存不如 HTTP 高效，会增加数据开销和功耗，甚至已有的安全措施也会因此而受到影响；

(3) SSL 证书需要钱，功能越强大的证书费用越高，个人网站、小网站没有必要一般不会用。

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

(4) SSL 证书通常需要绑定 IP，不能在同一 IP 上绑定多个域名，IPv4 资源不可能支撑这个消耗。

(5) HTTPS 协议的加密范围也比较有限，在黑客攻击、拒绝服务攻击、服务器劫持等方面几乎起不到什么作用。最关键的，SSL 证书的信用链体系并不安全，特别是在某些国家可以控制 CA 根证书的情况下，中间人攻击一样可行。

六、http 切换到 HTTPS

如果需要将网站从 http 切换到 https 到底该如何实现呢？

这里需要将页面中所有的链接，例如 js，css，图片等等链接都由 http 改为 https。例如：baidu.com 改为 https://baidu.com

BTW，这里虽然将 http 切换为了 https，还是建议保留 http。所以我们在切换的时候可以做 http 和 https 的兼容，具体实现方式是，去掉页面链接中的 http 头部，这样可以自动匹配 http 头和 https 头。例如：将 http://baidu.com 改为 //baidu.com。然后当用户从 http 的入口进入访问页面时，页面就是 http，如果用户是从 https 的入口进入访问页面，页面即使 https 的。

七、提到 UI 级别测试和 API 测试之间的关键区别？



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

UI（用户界面）是指测试图形界面，如用户如何与应用程序交互，测试应用程序元素，如字体，图像，布局等。UI 测试基本上侧重于应用程序的外观和感觉。

而 API 可以实现两个独立的软件系统之间的通信。实现 API 的软件系统包含可由另一软件系统执行的功能或子例程

请详细阐述接口测试和 UI 测试在测试活动中是如何协同测试的？

八、接口测试用例的编写要点有哪些？

- 1) 必填字段：请求参数必填项、可选项
- 2) 合法性：输入输出合法、非法参数
- 3) 边界：请求参数边界值等
- 4) 容错能力：大容量数据、频繁请求、重复请求（如：订单）、异常网络等的处理
- 5) 响应数据校验：断言、数据提取传递到下一级接口...
- 6) 逻辑校验：如两个请求的接口有严格的先后顺序，需要测试调转顺序的情况



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

7) 性能：对接口模拟并发测试，逐步加压，分析瓶颈点

8) 安全性：构造恶意的字符请求，如：SQL 注入、XSS、敏感信息、业务逻辑（如：跳过某些关键步骤；未经验证操纵敏感数据）

- * 测试每个参数类型不合法的情况(类型不合法容易遗漏 NULL 型)
- * 测试每个参数取值范围不合法的情况
- * 测试参数为空的情况
- * 测试参数前后台定义的一致性
- * 测试每个参数的上下限(这里容易出致命的 BUG, 如果程序处理不当，可能导致崩溃)
- * 如果两个请求有严格的先后顺序，需要测试调转顺序的情况

九、接口自动化测试的流程？

基本的接口功能自动化测试流程为：需求分析-->用例设计-->脚本开发-->测试执行-->结果分析

十、POST 和 GET 有什么区别？

答案

GET 在浏览器回退时是无害的，而 POST 会再次提交请求。



更多软件测试资料可以关注公众号：**软件测试资料侠**

GET 产生的 URL 地址可以被**保存为书签**，而 POST 不可以。

GET 请求会被浏览器**主动 cache**，而 POST 不会，除非手动设置。

GET 请求只能进行 url 编码，而 **POST 支持多种编码方式**。

GET 请求**参数**会被完整保留在**浏览器历史记录**里，而 POST 中的参数不会被保留。

GET 请求在 URL 中传送的参数是**有长度限制**的，而 POST 没有。

GET 比 POST 更不安全，因为参数直接暴露在 URL 上，所以不能用来传递敏感信息。

GET 参数通过 **URL 传递**，POST 放在请求 **Body** 中。

GET 产生一个 **TCP 数据包**，POST 产生**两个 TCP 数据包**。

GET 请求的 URL 传参有长度限制，而 POST 请求没有长度限制

GET 请求的参数只能是 ASCII 码，所以中文需要 URL 编码，而 POST 请求传参没有这个限制；

GET 产生一个 TCP 数据包；POST 产生两个 TCP 数据包

十一、常见的 HTTP Header 及其作用

答案

常规 HTTP 头

Request URL: jintix.com/

更多软件测试资料可以关注公众号：[软件测试资料侠](#)



柠檬班

让学习变得简单

Request Method: GET

Status Code: 200 OK

Remote Address: 14.215.177.38:443

相应头:

Cache-Control: private

Connection: Keep-Alive

Content-Encoding: gzip

Content-Type: text/html; charset=utf-8

Date: Wed, 27 Mar 2019 09:18:39 GMT

Expires: Wed, 27 Mar 2019 09:18:39 GMT

Server: BWS/1.1

Set-Cookie:

H_PS_PSSID=26524_1449_21099_28721_28558_28697_28585
_28640_26350_28702; path=/

Strict-Transport-Security: max-age=172800



柠檬班

让学习变得简单

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

Transfer-Encoding: chunked

X-Ua-Compatible: IE=Edge,chrome=1

请求头:

Accept:

text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8

Accept-Encoding: gzip, deflate, br

Accept-Language: zh-CN,zh;q=0.9,en;q=0.8

Cache-Control: max-age=0

Connection: keep-alive

Cookie: BIDUPSID=8B429E47FD1DA3ED78C8389452B0E8C2;
PSTM=1529386822

Host: baidu.com

User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/68.0.3440.84 Safari/537.36



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

十二、Session 与 Cookie 有什么区别？

答案

保存位置。SESSION 数据保存在**服务器端**，Cookie 数据保存在**客户端**浏览器。

保存方式。SESSION 默认被存在在服务器的一个**文件**里，可以手动设置放在文件、数据库、或内存中；Cookie 默认保存在客户端**内存中**，如果设置了过期时间就保存在硬盘中。

依赖关系。SESSION 依赖 **Cookie** 来识别 session_id，如果浏览器禁用了 Cookie，SESSION 也会失效，此时可以**通过 url** 传递 session_id。

安全性。因为 SESSION 数据保存在服务端，所以 **SESSION 安全性比 Cookie 高**。

尺寸大小。SESSION 基本上**没有大小限制**，COOKIE 保存的**内容比较小**，具体由浏览器决定。

服务器性能。SESSION 对**服务器的压力会更大一些**，而 Cookie 放在客户端，所以对服务器基本没影响。

十三、HTTP 通讯过程中，是客户端还是服务端主动断开连接？

答案

不考虑 keepalive 的情况下，



更多软件测试资料可以关注公众号：**软件测试资料侠**

http1.0 带 content-length, body 长度可知, 客户端在接收 body 时, 就可以依据这个长度来接受数据。接受完毕后, 就表示这个请求完毕了。**客户端主动**调用 close 进入四次挥手。

不带 content-length, body 长度不可知, 客户端一直接受数据, 直到**服务端主动断开** http1.1 带 content-length body 长度可知 **客户端主动断开**带 Transfer-encoding: chunked body 会被分成多个块, 每块的开始会标识出当前块的长度, body 就不需要通过 content-length 来指定了。但依然可以知道 body 的长度 **客户端主动断开**

不带 Transfer-encoding: chunked 且不带 content-length 客户端接收数据, 直到**服务端主动断开连接**。

即：**如果能够有办法知道服务器传来的长度, 都是客户端首先断开。如果不知道就一直接收数据。知道服务端断开。**

十四、HTTP 有哪些请求方法?

HTTP 共有如下 7 种请求方式, 每种都可以发送 Header 和 Body:

GET

POST



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

PUT

DELETE

OPTIONS

HEAD

PATCH

十五、Cookie 保存在哪里？

答案

如果**设置了**过期时间，Cookie 保存在**硬盘**中。

如果**没有设置**过期时间，Cookie 保存在**内存**中。

十六、简述 TCP/IP 的三次握手和四次挥手

答案

三次握手

起初，服务器和客户端都为 CLOSED 状态。在通信开始前，双方都得创建各自的传输控制块（TCB）。服务器创建完 TCB 后遍进入 LISTEN 状态，此时准备接收客户端发来的连接请求。



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

第一次握手

客户端向服务端发送连接请求报文段。该报文段的头部中 $\text{SYN}=1$, $\text{ACK}=0$, $\text{seq}=x$ 。请求发送后，客户端便进入 SYN-SENT 状态。

PS1: $\text{SYN}=1$, $\text{ACK}=0$ 表示该报文段为连接请求报文。

PS2: x 为本次 TCP 通信的字节流的初始序号。

TCP 规定: $\text{SYN}=1$ 的报文段不能有数据部分,但要消耗掉一个序号。

第二次握手 服务端收到连接请求报文段后，如果同意连接，则会发送一个应答: $\text{SYN}=1$, $\text{ACK}=1$, $\text{seq}=y$, $\text{ack}=x+1$ 。该应答发送完成后便进入 SYN-RCVD 状态。

PS1: $\text{SYN}=1$, $\text{ACK}=1$ 表示该报文段为连接同意的应答报文。

PS2: $\text{seq}=y$ 表示服务端作为发送者时，发送字节流的初始序号。

PS3: $\text{ack}=x+1$ 表示服务端希望下一个数据报发送序号从 $x+1$ 开始的字节。

第三次握手

当客户端收到连接同意的应答后，还要向服务端发送一个确认报文段，表示：服务端发来的连接同意应答已经成功收到。该报文段的头部为: $\text{ACK}=1$, $\text{seq}=x+1$, $\text{ack}=y+1$ 。



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

客户端发完这个报文段后便进入 ESTABLISHED 状态，服务端收到这个应答后也进入 ESTABLISHED 状态，此时连接的建立完成！

四次挥手

TCP 连接的释放一共需要四步，因此称为『四次挥手』。

我们知道，TCP 连接是双向的，因此在四次挥手中，前两次挥手用于断开一个方向的连接，后两次挥手用于断开另一方向的连接。

第一次挥手

若 A 认为数据发送完成，则它需要向 B 发送连接释放请求。该请求只有报文头，头中携带的主要参数为：FIN=1，seq=u。此时，A 将进入 FIN-WAIT-1 状态。

PS1：FIN=1 表示该报文段是一个连接释放请求。

PS2：seq=u，u-1 是 A 向 B 发送的最后一个字节的序号。

第二次挥手

B 收到连接释放请求后，会通知相应的应用程序，告诉它 A 向 B 这个方向的连接已经释放。此时 B 进入 CLOSE-WAIT 状态，并向 A 发送连接释放的应答，其报文头包含：ACK=1，seq=v，ack=u+1。

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

PS1: ACK=1: 除 TCP 连接请求报文段以外, TCP 通信过程中所有数据报的 ACK 都为 1, 表示应答。

PS2: seq=v, v-1 是 B 向 A 发送的最后一个字节的序号。

PS3: ack=u+1 表示希望收到从第 u+1 个字节开始的报文段, 并且已经成功接收了前 u 个字节。

A 收到该应答, 进入 FIN-WAIT-2 状态, 等待 B 发送连接释放请求。

第二次挥手完成后, A 到 B 方向的连接已经释放, B 不会再接收数据, A 也不会再发送数据。但 B 到 A 方向的连接仍然存在, B 可以继续向 A 发送数据。

第三次挥手

当 B 向 A 发完所有数据后, 向 A 发送连接释放请求, 请求头: FIN=1, ACK=1, seq=w, ack=u+1。B 便进入 LAST-ACK 状态。

第四次挥手

A 收到释放请求后, 向 B 发送确认应答, 此时 A 进入 TIME-WAIT 状态。

该状态会持续 2MSL 时间, 若该时间段内没有 B 的重发请求的话, 就进入 CLOSED 状态, 撤销 TCB。

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

当 B 收到确认应答后，也便进入 CLOSED 状态，撤销 TCB。

为什么 A 要先进入 TIME-WAIT 状态，等待 2MSL 时间后才进入 CLOSED 状态？为了保证 B 能收到 A 的确认应答。若 A 发完确认应答后直接进入 CLOSED 状态，那么如果该应答丢失，B 等待超时后就会重新发送连接释放请求，但此时 A 已经关闭了，不会作出任何响应，因此 B 永远无法正常关闭。

十七、TCP 和 UDP 有什么区别

答案

TCP 面向连接（如打电话要先拨号建立连接）；UDP 是无连接的，即发送数据之前不需要建立连接

TCP 提供可靠的服务。也就是说，通过 TCP 连接传送的数据，无差错，不丢失，不重复，且按序到达；UDP 尽最大努力交付，即不保证可靠交付。

TCP 面向字节流，实际上是 TCP 把数据看成一连串无结构的字节流；UDP 是面向报文的 UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

每一条 TCP 连接只能是点到点的;UDP 支持一对一，一对多，多对一和多对多的交互通信

TCP 首部开销 20 字节;UDP 的首部开销小，只有 8 个字节

TCP 的逻辑通信信道是全双工的可靠信道，UDP 则是不可靠信道

十八、什么是 TCP/IP?

答案

TCP/IP，也就是互联网协议套件（英语：Internet Protocol Suite，缩写 IPS）是一个**网络通信模型**，以及一整个**网络传输协议家族**，为网际网络的基础通信架构。

因为该协议家族的两个核心协议：TCP（传输控制协议）和 IP（网际协议）为该家族中最早通过的标准，所以它常被通称为 **TCP/IP 协议族**，简称 TCP/IP。

由于在网络通讯协议普遍采用分层的结构，当多个层次的协议共同工作时，类似计算机科学中的堆栈，因此又被称为 **TCP/IP 协议栈**。

TCP/IP 提供点对点的链接机制，将数据应该如何封装、定址、传输、路由以及在目的地如何接收，都加以标准化。

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

它将软件通信过程抽象化为**四个抽象层**，采取协议堆栈的方式，分别实现出不同通信协议。

协议族下的各种协议，依其功能不同，被分别归属到这四个层次结构之中，常被视为是简化的**七层 OSI 模型**。

十九、json 和字典 dict 的区别？

现在自动化培训烂大街，是个人都能说的上几个框架，面试如果问框架相关问题，求职者只需一瓶 82 年的雪碧，会吹的让你怀疑人生！

所以面试官为了更清楚的知道你是停留在表面上的花拳绣腿还是有扎实的基础，就不会问框架这种东西了。基本上问几个数据类型的基础就知道有没货了。

那么 json 和字典到底有什么区别呢？初学者连 python 的基础数据类型都没搞清楚，直接撸框架，有的人学了几个月可能都迷迷糊糊的，以为 json 就是字典。这个是肯定不对的。

首先 python 里面的基础数据类型有：int、str、float、list、bool、tuple、dict、set 这几种类型，里面没 json 这种数据类型。

JSON(JavaScript Object Notation, JS 对象简谱) 是一种轻量级的数据交换格式。它基于 ECMAScript (欧洲计算机协会制定的 js



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。

由于你的代码是 python 写的（也有可能是 php,java,c, ruby 等语言），但是后端接口是 java 写的（也有可能是其它语言），不同的语言数据类型是不一样的（就好比中国的语言和美国的语言数据类型也不一样，中国的一般说一只羊，一头牛，美国都是 a /an 这种单位），所以就导致你提交的数据，别的开发语言无法识别，这就需要规范传输的数据（传输的数据都是一个字符串），大家都遵循一个规范，按一个标准的格式去传输，于是就有 json 这种国际化规范的数据类型。

json 本质上还是字符串，只是按 key: value 这种键值对的格式来的字符串

```
import json
```

```
# a 是字典 dict
```

```
a = {"a": 1, "b": 2, "c": True}
```

```
# b 是 json
```



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

```
b = '{"a": 1, "b": 2, "c": true}'
```

```
print(type(a))
```

```
print(json.dumps(a)) # a 转 json
```

运行结果

```
<class 'dict'>
```

```
{"a": 1, "b": 2, "c": true}
```

```
<class 'str'>
```

```
{'a': 1, 'b': 2, 'c': True}
```

二十、测试的数据你放在哪？

测试数据到底该怎么放，这个是面试官最喜欢问的一个题了，似乎仁者见仁智者见智，没有标准的答案，有的人说放 excel，也有的说放.py 脚本，也有的说放 ini 配置文件，

还有放到 json，yaml 文件，txt 文件，甚至有的放数据库，五花八门，一百个做自动化的小伙伴有 100 个放的地方。

这里总结下测试的数据到底该怎么放？



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

首先测试的数据是分很多种的，有登录的账户数据，也有注册的账户数据，还有接口的参数，还有邮箱配置的数据等等等等，所以这个题不能一概而论给答死了。要不然就是给自己挖坑。

以下两个大忌不能回答：

测试的数据是不能写死到代码里面的，这个是原则问题，也是写代码的大忌（你要是回答写在代码里面，估计就是回去等通知了）

测试数据放到.py 的开头，这种其实很方便，对于少量的，固定不变的数据其实是可以放的，但是面试时候，千万不能这样说，面试官喜欢装逼的方法

测试数据存放总结：

- 1.对于账号密码，这种管全局的参数，可以用命令行参数，单独抽出来，写的配置文件里（如 ini）
- 2.对于一些一次性消耗的数据，比如注册，每次注册不一样的数，可以用随机函数生成
- 3.对于一个接口有多组测试的参数，可以参数化，数据放yaml,text,json,excel 都可以
- 4.对于可以反复使用的数据，比如订单的各种状态需要造数据的情况，可以放到数据库，每次数据初始化，用完后再清理



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

5.对于邮箱配置的一些参数，可以用 ini 配置文件

6.对于全部是独立的接口项目，可以用数据驱动方式，用 excel/csv 管理测试的接口数据

7.对于少量的静态数据，比如一个接口的测试数据，也就 2-3 组，可以写到 py 脚本的开头，十年八年都不会变更的

总之不同的测试数据，可以用不同的文件管理

二十一、什么是数据驱动，如何参数化？

参数化和数据驱动的概念这个肯定要知道的，参数化的思想是代码用例写好了后，不需要改代码，只需维护测试数据就可以了，并且根据不同的测试数据生成多个用例

二十二、下个接口请求参数依赖上个接口的返回数据

这个很容易，不同的接口封装成不同的函数或方法，需要的数据 return 出来，用一个中间变量 a 去接受，后面的接口传 a 就可以了



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

二十三、依赖于登录的接口如何处理

登录接口依赖 token 的，可以先登录后，token 存到一个 yaml 或者 json，或者 ini 的配置文件里面，后面所有的请求去拿这个数据就可以全局使用了。

如果是 cookies 的参数，可以用 session 自动关联

```
s=requests.session()
```

后面请求用 s.get()和 s.post()就可以自动关联 cookies 了

二十四、依赖第三方的接口如何处理

这个需要自己去搭建一个 mock 服务，模拟接口返回数据，mock 服务搭建需要自己能够熟练掌握，面试会问你具体如何搭建，如何模拟返回的数据，是用的什么格式，如何请求的。

二十五、不可逆的操作，如何处理，比如删除一个订单这种接口如何测试

此题考的是造数据的能力，接口的请求数据，很多都是需要依赖前面一个状态的。



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

比如工作流这种，流向不同的人状态不一样，操作权限不一样，测试的时候，每种状态都要测到,就需要自己会造数据了。

平常手工测试造数据，直接在数据库改字段状态。那么自动化也是一样，造数据可以用 python 连数据库了，做增删改查的操作。

测试用例前置操作，setUp 做数据准备

后置操作，tearDown 做数据清理

二十六、接口产生的垃圾数据如何清理

跟上面一样，造数据和数据清理，需用 python 连数据库了，做增删改查的操作

测试用例前置操作，setUp 做数据准备

后置操作，tearDown 做数据清理

二十七、一个订单的几种状态如何全部测到，如：未处理，处理中，处理失败，处理成功

跟上面一样，也是考察造数据，修改数据的状态



二十八、python 如何连接数据库操作？

这个就是详细的考察你是如何用 python 连数据库的，并且最好能现场写代码那种（有的笔试题就是 python 连数据库）

具体问你用到哪个模块，查询的数据是什么类型？如何删除数据？如何新增数据？如何修改数据？

PyMySQL 是在 Python3.x 版本中用于连接 MySQL 服务器的一个库，Python2 中则使用 mysqlldb。

二十九、在 API 测试中测试的常用协议是什么？

HTTP

JMS

REST

SOAP

UDDI



三十、接口测试的步骤有哪些？

- 1) 发送接口请求
- 2) 测试接口获取返回值
- 3) 断言：判断实际结果是否符合预期

三十一、接口测试中依赖登录状态的接口如何测试？

依赖登录状态的接口，本质上是在每次发送请求时需要带上存储有账户有效信息的 Session 或 Cookie 才能发送成功，在构建 POST 请求时添加必要的 Session 或 Cookie

三十二、依赖于第三方数据的接口如何进行测试？

可以利用一些 MOCK 工具（如：JSON Server、Easy Mock）来模拟第三方的数据返回，最大限度的降低对第三方数据接口的依赖

三十三、解释什么是 SOAP？



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

SOAP 代表简单对象访问控制，它是一种基于 XML 的协议，用于在计算机之间交换信息。

三十四、解释什么是 REST API?

这是开发人员执行请求并接收响应的一组功能。在 REST API 中，通过 HTTP 协议进行交互

REST - 代表状态转移，它正快速成为 API 创建的标准。

三十五、API Builder 如何工作?

API Builder 是一个由四个 SQL 文件组成的 PLSQL 程序

要设置 API 参数并启动该过程，一个文件是负责的

为临时表和主包创建两个文件以创建输出的代码

第四个文件将代码的“假脱机”输出创建一个名为“output_script.sql”的文件

三十六、解释什么是 TestApi?



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

TestApi 是一个实用程序和测试 API 库，可让测试人员和开发人员为 .NET 和 Win32 应用程序创建测试工具和自动测试。它提供了一套常见的测试构建块，类型，数据结构和算法。

三十七、什么是使用 runscope 的 API 测试？

Runscope 是一个 Web 应用程序，它提供后端服务和易于使用的界面来测试 API。

三十八、解释 API 测试设计的原理是什么？

API 测试设计的原理是

安装：创建对象，启动服务，初始化数据等

执行：执行 API 或场景的步骤，也记录日志

验证：比较预期结果和实际结果

报告：查看执行情况，通过，失败或阻止

清理：回到最初的测试状态



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

三十九、API 测试发现的 Bug 类型是什么？

缺少或重复的功能

无法正常处理错误条件

强调

可靠性

安全

未使用的标志

未实现错误

错误处理不一致

性能

多线程问题

错误不正确

我们测试的接口属于哪一类？

服务器接口（基于 HTTP 协议的接口）



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

大多数人常说的接口测试，通常是 B/S 架构，由客户端（浏览器）调用，或模拟客户端（浏览器）调用服务器提供的请求接口，由服务器完成处理并返回一个应答的过程。

例如：Webservice 接口，http 接口，jms 接口，hessian 接口。

四十、TCP 长连接和短连接的区别

当网络通信时采用 TCP 协议时，在真正的读写操作之前，server 与 client 之间必须建立一个连接，当读写操作完成后，双方不再需要这个连接时它们可以释放这个连接，连接的建立是需要三次握手的，而释放则需要 4 次挥手，所以说每个连接的建立都是需要资源消耗和时间消耗的

长连接：

所谓长连接，指在一个 TCP 连接上可以连续发送多个数据包，在 TCP 连接保持期间，如果没有数据包发送，需要双方发检测包以维持此连接，一般需要自己做在线维持（不发生 RST 包和四次挥手）。

连接→数据传输→保持连接(心跳)→数据传输→保持连接(心跳)
→.....→关闭连接（一个 TCP 连接通道多个读写通信）；



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

这就要求长连接在没有数据通信时，定时发送数据包(心跳)，以维持连接状态；

TCP 保活功能，保活功能主要为服务器应用提供，服务器应用希望知道客户主机是否崩溃，从而可以代表客户使用资源。如果客户已经消失，使得服务器上保留一个半开放的连接，而服务器又在等待来自客户端的数据，则服务器将应远等待客户端的数据，保活功能就是试图在服务器端检测到这种半开放的连接。

如果一个给定的连接在两小时内没有任何的动作，则服务器就向客户发一个探测报文段，客户主机必须处于以下 4 个状态之一：

客户主机依然正常运行，并从服务器可达。客户的 TCP 响应正常，而服务器也知道对方是正常的，服务器在两小时后将保活定时器复位。

客户主机已经崩溃，并且关闭或者正在重新启动。在任何一种情况下，客户的 TCP 都没有响应。服务端将不能收到对探测的响应，并在 75 秒后超时。服务器总共发送 10 个这样的探测，每个间隔 75 秒。如果服务器没有收到一个响应，它就认为客户主机已经关闭并终止连接。

客户主机崩溃并已经重新启动。服务器将收到一个对其保活探测的响应，这个响应是一个复位，使得服务器终止这个连接。

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

客户机正常运行，但是服务器不可达，这种情况与 2 类似，TCP 能发现的就是没有收到探查的响应。

短连接：

短连接是指通信双方有数据交互时，就建立一个 TCP 连接，数据发送完成后，则断开此 TCP 连接（管理起来比较简单，存在的连接都是有用的连接，不需要额外的控制手段）；

连接→数据传输→关闭连接；

应用场景：

长连接多用于操作频繁（读写），点对点的通讯，而且连接数不能太多情况，。每个 TCP 连接都需要三步握手，这需要时间，如果每个操作都是先连接，再操作的话那么处理速度会降低很多，所以每个操作完后都不断开，次处理时直接发送数据包就 OK 了，不用建立 TCP 连接。例如：数据库的连接用长连接， 如果用短连接频繁的通信会造成 socket 错误，而且频繁的 socket 创建也是对资源的浪费。

而像 WEB 网站的 http 服务一般都用短链接（http1.0 只支持短连接，1.1keep alive 带时间，操作次数限制的长连接），因为长连接对于服务端来说会耗费一定的资源，而像 WEB 网站这么频繁的成千



柠檬班

让学习变得简单

更多软件测试资料可以关注公众号：[软件测试资料侠](#)

上万甚至上亿客户端的连接用短连接会更省一些资源，如果用长连接，而且同时有成千上万的用户，如果每个用户都占用一个连接的话，那可想而知吧。所以并发量大，但每个用户无需频繁操作情况下需用短连好；

在长连接中一般是没有条件能够判断读写什么时候结束，所以必须要加长度报文头。读函数先是读取报文头的长度，再根据这个长度去读相应长度的报文。

四十一、Session 和 cookie 的区别

cookie

cookie 的原理和作用

cookie 的原理

什么是 cookie

分类两种

内存 cookie



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

相关的 cookie 信息存放在浏览器的缓存中

如果浏览器关闭后，cookie 信息被释放

硬盘 cookie

cookie 信息是保存在客户端机器的硬盘中

永久保存，可以反复读取，除非用户执行删除操作

易用性好

安全性差

四十二、为什么要有 cookie，它的作用是什么？

cookie 有什么作用？

cookie 可以解决 http 的无状态的问题，与服务器进行交互，作为 http 规范存在。它具有极高的简便性、可扩展性和可用性，也可以通过加密和 SSL 技术来提高其安全性。因此推荐使用 cookie 作为标识而不是身份验证的工具。

其中 cookie 的作用就是为了解决 HTTP 协议无状态的缺陷所作出的努力



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

cookie 测试点

四十三、Cookie 测试的测试点

1.禁止使用 Cookie

设置浏览器禁止使用 Cookie，访问网页后，检查存放 Cookie 文件中未生成相关文件；

2.Cookie 存储路径

按照操作系统和浏览器对 Cookie 存放路径的设置，检查存放路径是否与设置一致；

3.Cookie 过期检查

按照 Cookie 过期时间，检查存放文件该 Cookie 是否被自动删除

4.检查浏览器中 Cookie 选项

通过不同浏览器，设置是否接受 Cookie 文件，如同意接受 Cookie，检查存放路径中是否存在 Cookie 文件

5.浏览器删除 Cookie

通过浏览器的设置，删除 Cookie 文件



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

6.Cookie 加密

提交敏感信息时，数据应加密

7.Cookie 保存信息

验证 Cookie 能正常工作

8.篡改 Cookie

修改 Cookie 内容，查看系统功能是否出现异常，或数据错乱

9.Cookie 的兼容性

使用不同类型，或同一类型不同版本的浏览器，检查 cookie 文件的兼容性

10.刷新操作对 cookie 的影响

进行刷新操作后，是否重新生成 cookie 文件或是对 cookie 文件进行修改

11.检查 cookie 内容存储是否完整正确

若 cookie 进行了加密，先对 cookie 文件内容进行解密，然后检查是否按照设计要求存储了相关所有的 cookie 记录信息。



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

12.对应硬盘存储空间没有空闲时，是否能进行 cookie 内容的有效存储

13.多次做相同的操作或设置，检查是否更新或添加了新的 cookie 文件

按照设计要求进行判断

14.如果使用 cookie 来统计次数，则要检测是否统计正确

例如通过用户登录次数进行统计

1,访问网站，每次都需要输入用户名和密码

分第一次登陆还是非第一次登录

四十四、cookie 的缺点

(1) 大小和数目受限制。浏览器对一个域 cookie 的条目数有上限要求，且每个 cookie 的大小不得超过 4kb。

(2) 存在安全性问题，易被人拦截。

(3) 需要指定域，不可以跨域



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

(4) 浪费带宽，因为我每次请求一个新的页面，cookie 都会被自动发送过去。

(5) 有的移动端浏览器不支持 cookie 或浏览器禁用 cookie

(6) 有些状态不可能保存在客户端。例如，为了防止重复提交表单，我们需要在服务器端保存一个计数器。如果我们把这个计数器保存在客户端，那么它起不到任何作用。

四十五、session 的作用

1, 为什么要用 cookie

a:方便用户进行登录

b: 收集用户的相关登录时间，时长，使用习惯

2, 为什么要使用 session

s1: 登录网站后

s2: 跳转到其他页面

其他协议

无状态协议



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

不记录用户身份信息

产生的问题

1, 只要

2, 任意人都

s3: 产生了 session 技术处理

3, session 的原理

1: 当用户注册后

2, 在本地生成 cookie 文件, 保留用户信息

3, 用户登录

1, 系统检查本地是否存在 cookie 文件

若不存在, 等待登录后,

在本地创建 cookie 文件

2, 若已存在, 系统在用户登录的同时, 由服务器端口自动随机分配一个 sessionID 传入 cookie 文件



更多软件测试资料可以关注公众号：[软件测试资料侠](#)

4，若用户打开某个需要权限设置的页面，系统回去检查对应的 sessionID 是否存在有效正确

session 机制则是又一种在客户端与服务器之间保持状态的解决方案。

四十六、cookie 与 session 的区别：

1、cookie 数据存放在客户的浏览器上，session 数据放在服务器上。

2、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。

3、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。

4、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。

5、所以个人建议：

将登陆信息等重要信息存放为 session

其他信息如果需要保留，可以放在 cookie 中



更多软件测试资料可以关注公众号：[软件测试资料侠](#)
作业

知道 cookie 和 session 的关系

知道 cookie 是如何产生的

知道 sessionid 的产生和作用

知道 cookie 上的重要字段和作用

看完中高级测试工程师面试题，面试更有底气了！

柠檬班软件测试资料侠