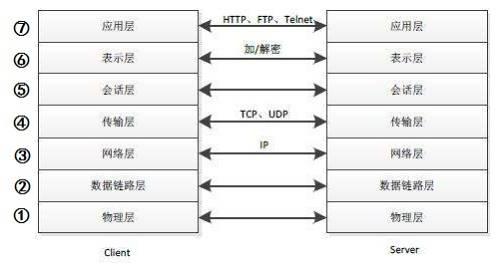
**、按你的理解，软件接口是什么？**  
**答：**  
就是指程序中具体负责在不同模块之间传输或接受数据的并做处理的类或者函数。

**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）；
* 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？**  
**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。

**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。

**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；

**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。