## 前端面试题（1）

### 1、说一下http 和https

参考回答：  
https 的SSL 加密是在传输层实现的。

#### (1)http 和https 的基本概念

http: 超文本传输协议，是互联网上应用最为广泛的一种网络协议，是一个客户端和服务器端请求和应答的标准（TCP），用于从WWW 服务器传输超文本到本地浏览器的传输协议，它可以使浏览器更加高效，使网络传输减少。  
 https: 是以安全为目标的HTTP 通道，简单讲是HTTP 的安全版，即HTTP 下加入SSL层，HTTPS 的安全基础是SSL，因此加密的详细内容就需要SSL。  
https 协议的主要作用是：建立一个信息安全通道，来确保数组的传输，确保网站的真实  
性。

#### (2)http 和https 的区别？

http 传输的数据都是未加密的，也就是明文的，网景公司设置了SSL 协议来对http 协议传输的数据进行加密处理，简单来说https 协议是由http 和ssl 协议构建的可进行加密传输和身份认证的网络协议，比http 协议的安全性更高。  
主要的区别如下：  
Https 协议需要ca 证书，费用较高。  
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。  
使用不同的链接方式，端口也不同，一般而言，http 协议的端口为80，https 的端口为  
443  
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。

#### (3)https 协议的工作原理

客户端在使用HTTPS 方式与Web 服务器通信时有以下几个步骤，如图所示。  
客户使用https url 访问服务器，则要求web 服务器建立ssl 链接。  
web 服务器接收到客户端的请求之后，会将网站的证书（证书中包含了公钥），返回或  
者说传输给客户端。  
客户端和web 服务器端开始协商SSL 链接的安全等级，也就是加密等级。  
客户端浏览器通过双方协商一致的安全等级，建立会话密钥，然后通过网站的公钥来加  
密会话密钥，并传送给网站。  
web 服务器通过自己的私钥解密出会话密钥。  
web 服务器通过会话密钥加密与客户端之间的通信。

#### (4)https 协议的优点

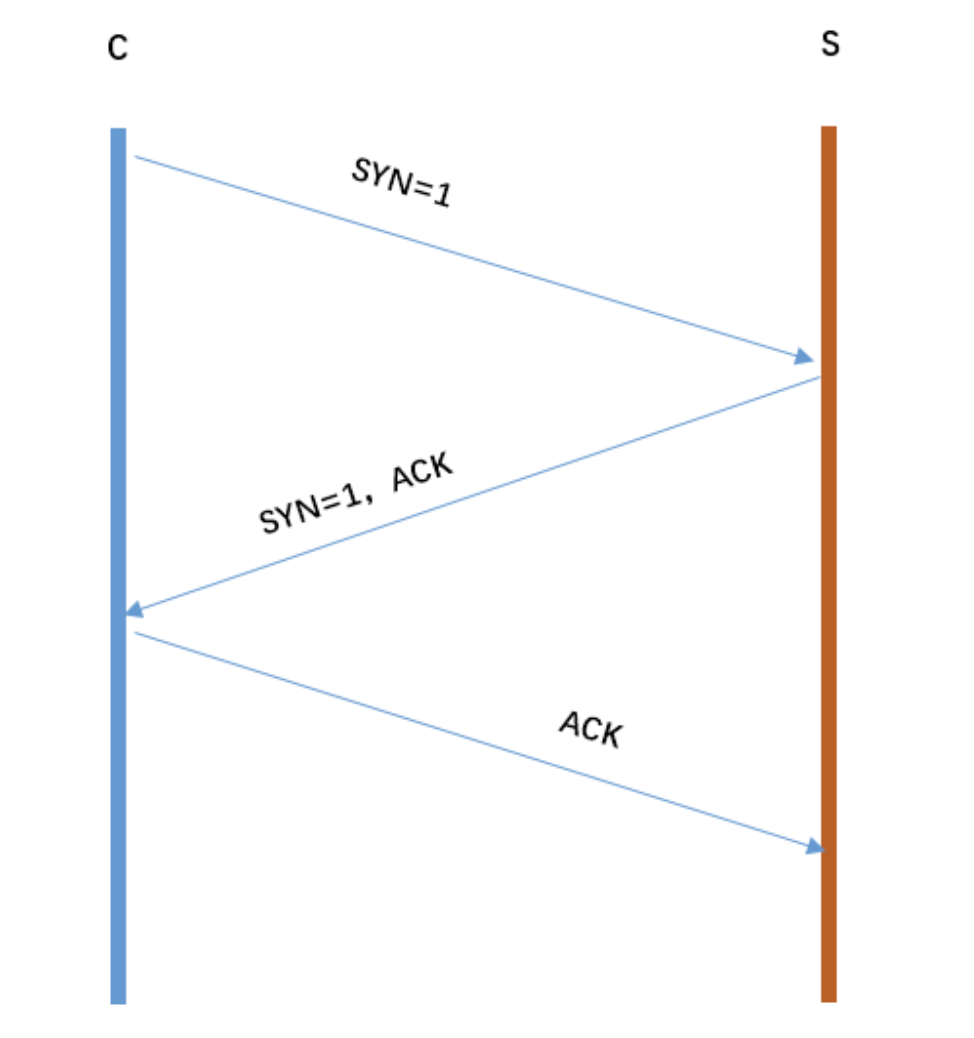
使用HTTPS 协议可认证用户和服务器，确保数据发送到正确的客户机和服务器；

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。  
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。  
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”。

##### (5)https 协议的缺点

https 握手阶段比较费时，会使页面加载时间延长50%，增加10%~20%的耗电。  
https 缓存不如http 高效，会增加数据开销。  
SSL 证书也需要钱，功能越强大的证书费用越高。  
SSL 证书需要绑定IP，不能再同一个ip 上绑定多个域名，ipv4 资源支持不了这种消耗。

### 2、tcp 三次握手，一句话概括

参考回答：  
客户端和服务端都需要直到各自可收发，因此需要三次握手。  
简化三次握手：  
  
 从图片可以得到三次握手可以简化为：C 发起请求连接S 确认，也发起连接C 确认我们  
再看看每次握手的作用：第一次握手：S 只可以确认自己可以接受C 发送的报文段第  
二次握手：C 可以确认S 收到了自己发送的报文段，并且可以确认自己可以接受S 发  
送的报文段第三次握手：S 可以确认C 收到了自己发送的报文段

### 3、TCP 和UDP 的区别

参考回答：  
（1）TCP 是面向连接的，udp 是无连接的即发送数据前不需要先建立链接。  
（2）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。并且因为tcp 可靠，  
面向连接，不会丢失数据因此适合大数据量的交换。  
（3）TCP 是面向字节流，UDP 面向报文，并且网络出现拥塞不会使得发送速率降低（因  
此会出现丢包，对实时的应用比如IP 电话和视频会议等）。  
（4）TCP 只能是1 对1 的，UDP 支持1 对1,1 对多。  
（5）TCP 的首部较大为20 字节，而UDP 只有8 字节。  
（6）TCP 是面向连接的可靠性传输，而UDP 是不可靠的。

### 4、Web Socket的实现和应用

参考回答：  
(1)什么是WebSocket?

WebSocket 是HTML5 中的协议，支持持久连续，http 协议不支持持久性连接。Http1.0  
和HTTP1.1 都不支持持久性的链接，HTTP1.1 中的keep-alive，将多个http 请求合并为  
1 个  
(2)WebSocket 是什么样的协议，具体有什么优点？  
HTTP 的生命周期通过Request 来界定，也就是Request 一个Response，那么在Http1.0  
协议中，这次Http 请求就结束了。在Http1.1 中进行了改进，是的有一个connection：  
Keep-alive，也就是说，在一个Http 连接中，可以发送多个Request，接收多个Response。  
但是必须记住，在Http 中一个Request 只能对应有一个Response，而且这个Response  
是被动的，不能主动发起。  
WebSocket 是基于Http 协议的，或者说借用了Http 协议来完成一部分握手，在握手阶段  
与Http 是相同的。我们来看一个websocket 握手协议的实现，基本是2 个属性，upgrade，  
connection。  
基本请求如下：  
GET /chat HTTP/1.1  
Host: server.example.com  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==  
Sec-WebSocket-Protocol: chat, superchat  
Sec-WebSocket-Version: 13  
Origin: http://example.com  
多了下面2 个属性：  
1 .Upgrade:webSocket

2.Connection:Upgrade  
告诉服务器发送的是websocket  
1.Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==  
2.Sec-WebSocket-Protocol: chat, superchat  
3.Sec-WebSocket-Version: 13

### 5、HTTP 请求的方式，HEAD 方式

参考回答：  
head：类似于get 请求，只不过返回的响应中没有具体的内容，用户获取报头  
options：允许客户端查看服务器的性能，比如说服务器支持的请求方式等等。

### 6、一个图片url 访问后直接下载怎样实现？

参考回答：  
请求的返回头里面，用于浏览器解析的重要参数就是OSS 的API 文档里面的返回http  
头，决定用户下载行为的参数。  
下载的情况下：

1. x-oss-object-type:  
   Normal
2. x-oss-request-id:  
   598D5ED34F29D01FE2925F41
3. x-oss-storage-class:  
   Standard

### 7、说一下web Quality（无障碍）

参考回答：  
能够被残障人士使用的网站才能称得上一个易用的（易访问的）网站。  
残障人士指的是那些带有残疾或者身体不健康的用户。  
使用alt 属性：  
this is a person  
有时候浏览器会无法显示图像。具体的原因有：  
用户关闭了图像显示  
浏览器是不支持图形显示的迷你浏览器  
浏览器是语音浏览器（供盲人和弱视人群使用）  
如果您使用了alt 属性，那么浏览器至少可以显示或读出有关图像的描述。

### 8、几个很实用的BOM 属性对象方法?

参考回答：  
什么是Bom? Bom 是浏览器对象。有哪些常用的Bom 属性呢？  
(1)location 对象  
location.href-- 返回或设置当前文档的URL  
location.search -- 返回URL 中的查询字符串部分。例  
如http://www.dreamdu.com/dreamdu.php?id=5&name=dreamdu 返回包括(?)后面的内  
容?id=5&name=dreamdu  
location.hash -- 返回URL#后面的内容，如果没有#，返回空  
location.host -- 返回URL 中的域名部分，例如www.dreamdu.com  
location.hostname -- 返回URL 中的主域名部分，例如dreamdu.com  
location.pathname -- 返回URL 的域名后的部分。例如http://www.dreamdu.com/xhtml/ 返  
回/xhtml/  
location.port -- 返回URL 中的端口部分。例如http://www.dreamdu.com:8080/xhtml/ 返回  
8080  
location.protocol -- 返回URL 中的协议部分。例如http://www.dreamdu.com:8080/xhtml/ 返  
回(//)前面的内容http:  
location.assign -- 设置当前文档的URL

location.replace() -- 设置当前文档的URL，并且在history 对象的地址列表中移除这个  
URL location.replace(url);  
location.reload() -- 重载当前页面  
(2)history 对象  
history.go() -- 前进或后退指定的页面数history.go(num);  
history.back() -- 后退一页  
history.forward() -- 前进一页  
(3)Navigator 对象  
navigator.userAgent -- 返回用户代理头的字符串表示(就是包括浏览器版本信息等的字  
符串)  
navigator.cookieEnabled -- 返回浏览器是否支持(启用)cookie

### 9、说一下http2.0

首先补充一下，http 和https 的区别，相比于http,https 是基于ssl 加密的http 协议  
简要概括：http2.0 是基于1999 年发布的http1.0 之后的首次更新。  
提升访问速度（可以对于，请求资源所需时间更少，访问速度更快，相比http1.0）  
允许多路复用：多路复用允许同时通过单一的HTTP/2 连接发送多重请求-响应信息。改  
善了：在http1.1 中，浏览器客户端在同一时间，针对同一域名下的请求有一定数量限  
制（连接数量），超过限制会被阻塞。  
二进制分帧：HTTP2.0 会将所有的传输信息分割为更小的信息或者帧，并对他们进行二  
进制编码  
首部压缩  
服务器端推送

### 10、补充400 和401、403 状态码

参考回答：  
(1)400 状态码：请求无效  
产生原因：前端提交数据的字段名称和字段类型与后台的实体没有保持一致  
前端提交到后台的数据应该是json 字符串类型，但是前端没有将对象JSON.stringify 转  
化成字符串。  
解决方法：  
对照字段的名称，保持一致性  
将obj 对象通过JSON.stringify 实现序列化  
(2)401 状态码：当前请求需要用户验证  
(3)403 状态码：服务器已经得到请求，但是拒绝执行

### 11、fetch 发送2 次请求的原因

fetch 发送post 请求的时候，总是发送2 次，第一次状态码是204，第二次才成功？  
原因很简单，因为你用fetch 的post 请求的时候，导致fetch 第一次发送了一个Options  
请求，询问服务器是否支持修改的请求头，如果服务器支持，则在第二次中发送真正的  
请求。

### 12、Cookie、sessionStorage、localStorage 的区别

共同点：都是保存在浏览器端，并且是同源的  
Cookie：cookie 数据始终在同源的http 请求中携带（即使不需要），即cookie 在浏览器  
和服务器间来回传递。而sessionStorage 和localStorage 不会自动把数据发给服务器，仅  
在本地保存。cookie 数据还有路径（path）的概念，可以限制cookie 只属于某个路径下,  
存储的大小很小只有4K 左右。（key：可以在浏览器和服务器端来回传递，存储容量  
小，只有大约4K 左右）  
sessionStorage：仅在当前浏览器窗口关闭前有效，自然也就不可能持久保持，localStorage：  
始终有效，窗口或浏览器关闭也一直保存，因此用作持久数据；cookie 只在设置的cookie  
过期时间之前一直有效，即使窗口或浏览器关闭。（key：本身就是一个回话过程，关  
闭浏览器后消失，session 为一个回话，当页面不同即使是同一页面打开两次，也被视为  
同一次回话）  
localStorage：localStorage 在所有同源窗口中都是共享的；cookie 也是在所有同源窗口中  
都是共享的。（key：同源窗口都会共享，并且不会失效，不管窗口或者浏览器关闭与  
否都会始终生效）  
补充说明一下cookie 的作用：  
保存用户登录状态。例如将用户id 存储于一个cookie 内，这样当用户下次访问该页面  
时就不需要重新登录了，现在很多论坛和社区都提供这样的功能。cookie 还可以设置  
过期时间，当超过时间期限后，cookie 就会自动消失。因此，系统往往可以提示用户保  
持登录状态的时间：常见选项有一个月、三个月、一年等。  
跟踪用户行为。例如一个天气预报网站，能够根据用户选择的地区显示当地的天气情况。  
如果每次都需要选择所在地是烦琐的，当利用了cookie 后就会显得很人性化了，系统能  
够记住上一次访问的地区，当下次再打开该页面时，它就会自动显示上次用户所在地区  
的天气情况。因为一切都是在后台完成，所以这样的页面就像为某个用户所定制的一样，使用起来非常方便定制页面。如果网站提供了换肤或更换布局的功能，那么可以使用cookie 来记录用户的选项，例如：背景色、分辨率等。当用户下次访问时，仍然可以  
保存上一次访问的界面风格。

### 13、iframe 是什么？有什么缺点？

定义：iframe 元素会创建包含另一个文档的内联框架  
提示：可以将提示文字放在之间，来提示某些不支持iframe 的浏览器  
缺点：  
会阻塞主页面的onload 事件  
搜索引擎无法解读这种页面，不利于SEO  
iframe 和主页面共享连接池，而浏览器对相同区域有限制所以会影响性能。

### 14、Cookie 如何防范XSS 攻击

XSS（跨站脚本攻击）是指攻击者在返回的HTML 中嵌入javascript 脚本，为了减轻这些  
攻击，需要在HTTP 头部配上，set-cookie：  
httponly-这个属性可以防止XSS,它会禁止javascript 脚本来访问cookie。  
secure - 这个属性告诉浏览器仅在请求为https 的时候发送cookie。  
结果应该是这样的：Set-Cookie=

### 15、介绍知道的http 返回的状态码

100 Continue 继续。客户端应继续其请求  
101 Switching Protocols 切换协议。服务器根据客户端的请求切换协议。只能切换到更  
高级的协议，例如，切换到HTTP 的新版本协议  
200 OK 请求成功。一般用于GET 与POST 请求  
201 Created 已创建。成功请求并创建了新的资源  
202 Accepted 已接受。已经接受请求，但未处理完成  
203 Non-Authoritative Information 非授权信息。请求成功。但返回的meta 信息不在原  
始的服务器，而是一个副本  
204 No Content 无内容。服务器成功处理，但未返回内容。在未更新网页的情况下，  
可确保浏览器继续显示当前文档

205 Reset Content 重置内容。服务器处理成功，用户终端（例如：浏览器）应重置文  
档视图。可通过此返回码清除浏览器的表单域  
206 Partial Content 部分内容。服务器成功处理了部分GET 请求  
300 Multiple Choices 多种选择。请求的资源可包括多个位置，相应可返回一个资源特  
征与地址的列表用于用户终端（例如：浏览器）选择  
301 Moved Permanently 永久移动。请求的资源已被永久的移动到新URI，返回信息会  
包括新的URI，浏览器会自动定向到新URI。今后任何新的请求都应使用新的URI 代替  
302 Found 临时移动。与301 类似。但资源只是临时被移动。客户端应继续使用原有  
URI  
303 See Other 查看其它地址。与301 类似。使用GET 和POST 请求查看  
304 Not Modified 未修改。所请求的资源未修改，服务器返回此状态码时，不会返回  
任何资源。客户端通常会缓存访问过的资源，通过提供一个头信息指出客户端希望只返  
回在指定日期之后修改的资源  
305 Use Proxy 使用代理。所请求的资源必须通过代理访问  
306 Unused 已经被废弃的HTTP 状态码  
307 Temporary Redirect 临时重定向。与302 类似。使用GET 请求重定向  
400 Bad Request 客户端请求的语法错误，服务器无法理解  
401 Unauthorized 请求要求用户的身份认证  
402 Payment Required 保留，将来使用  
403 Forbidden 服务器理解请求客户端的请求，但是拒绝执行此请求  
404 Not Found 服务器无法根据客户端的请求找到资源（网页）。通过此代码，网站  
设计人员可设置"您所请求的资源无法找到"的个性页面  
405 Method Not Allowed 客户端请求中的方法被禁止  
406 Not Acceptable 服务器无法根据客户端请求的内容特性完成请求  
407 Proxy Authentication Required 请求要求代理的身份认证，与401 类似，但请求者  
应当使用代理进行授权  
408 Request Time-out 服务器等待客户端发送的请求时间过长，超时  
409 Conflict 服务器完成客户端的PUT 请求是可能返回此代码，服务器处理请求时发  
生了冲突  
410 Gone 客户端请求的资源已经不存在。410 不同于404，如果资源以前有现在被永  
久删除了可使用410 代码，网站设计人员可通过301 代码指定资源的新位置  
411 Length Required 服务器无法处理客户端发送的不带Content-Length 的请求信息  
412 Precondition Failed 客户端请求信息的先决条件错误  
413 Request Entity Too Large 由于请求的实体过大，服务器无法处理，因此拒绝请求。  
为防止客户端的连续请求，服务器可能会关闭连接。如果只是服务器暂时无法处理，则  
会包含一个Retry-After 的响应信息  
414 Request-URI Too Large 请求的URI 过长（URI 通常为网址），服务器无法处理  
415 Unsupported Media Type 服务器无法处理请求附带的媒体格式  
416 Requested range not satisfiable 客户端请求的范围无效  
417 Expectation Failed 服务器无法满足Expect 的请求头信息  
500 Internal Server Error 服务器内部错误，无法完成请求  
501 Not Implemented 服务器不支持请求的功能，无法完成请求  
502 Bad Gateway 作为网关或者代理工作的服务器尝试执行请求时，从远程服务器接  
收到了一个无效的响应

503 Service Unavailable 由于超载或系统维护，服务器暂时的无法处理客户端的请求。  
延时的长度可包含在服务器的Retry-After 头信息中  
504 Gateway Time-out 充当网关或代理的服务器，未及时从远端服务器获取请求  
505 HTTP Version not supported 服务器不支持请求的HTTP 协议的版本，无法完成处  
理

### 16、讲讲304

304：如果客户端发送了一个带条件的GET 请求且该请求已被允许，而文档的内容（自  
上次访问以来或者根据请求的条件）并没有改变，则服务器应当返回这个304 状态码。

### 17、前端优化

降低请求量：合并资源，减少HTTP 请求数，minify / gzip 压缩，webP，lazyLoad。  
加快请求速度：预解析DNS，减少域名数，并行加载，CDN 分发。  
缓存：HTTP 协议缓存请求，离线缓存manifest，离线数据缓存localStorage。  
渲染：JS/CSS 优化，加载顺序，服务端渲染，pipeline。

### 18、GET 和POST 的区别

get 参数通过url 传递，post 放在request body 中。  
get 请求在url 中传递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 中，所以不能用来传递敏感信息。  
get 请求只能进行url 编码，而post 支持多种编码方式

get 请求会浏览器主动cache，而post 支持多种编码方式。  
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。  
GET 和POST 本质上就是TCP 链接，并无差别。但是由于HTTP 的规定和浏览器/服务器  
的限制，导致他们在应用过程中体现出一些不同。  
GET 产生一个TCP 数据包；POST 产生两个TCP 数据包。

### 19、HTTP 支持的方法

GET, POST, HEAD, OPTIONS, PUT, DELETE, TRACE, CONNECT