目录

[面试准备 1](#_Toc1536468427)

[网络相关 1](#_Toc2081481061)

[http和https的区别和特点 1](#_Toc943582597)

# 面试准备

# 网络相关

## http和https的区别和特点

http: 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超文本传输协议。

https:是以安全为目标的 HTTP 通道，即 HTTP 下 加入 SSL 层进行加密。其作用是：建立一个信息安全通道，来确保数据的传输，确保网站的真实性。

**区别：**

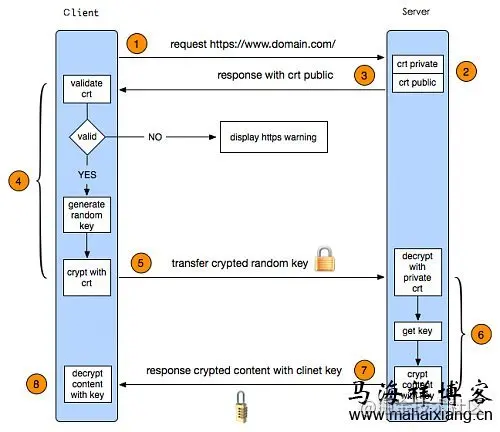
1、https协议需要到ca申请证书，一般免费证书较少，因而需要一定费用。

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。

## https的工作原理



1、客户端发起HTTPS请求

这个没什么好说的，就是用户在浏览器里输入一个https网址，然后连接到server的443端口。

2、服务端的配置

采用HTTPS协议的服务器必须要有一套数字证书，可以自己制作，也可以向组织申请，区别就是自己颁发的证书需要客户端验证通过，才可以继续访问，而使用受信任的公司申请的证书则不会弹出提示页面(startssl就是个不错的选择，有1年的免费服务)。

这套证书其实就是一对公钥和私钥，如果对公钥和私钥不太理解，可以想象成一把钥匙和一个锁头，只是全世界只有你一个人有这把钥匙，你可以把锁头给别人，别人可以用这个锁把重要的东西锁起来，然后发给你，因为只有你一个人有这把钥匙，所以只有你才能看到被这把锁锁起来的东西。

3、传送证书

这个证书其实就是公钥，只是包含了很多信息，如证书的颁发机构，过期时间等等。

4、客户端解析证书

这部分工作是有客户端的TLS来完成的，首先会验证公钥是否有效，比如颁发机构，过期时间等等，如果发现异常，则会弹出一个警告框，提示证书存在问题。

如果证书没有问题，那么就生成一个随机值，然后用证书对该随机值进行加密，就好像上面说的，把随机值用锁头锁起来，这样除非有钥匙，不然看不到被锁住的内容。

5、传送加密信息

这部分传送的是用证书加密后的随机值，目的就是让服务端得到这个随机值，以后客户端和服务端的通信就可以通过这个随机值来进行加密解密了。

6、服务段解密信息

服务端用私钥解密后，得到了客户端传过来的随机值(私钥)，然后把内容通过该值进行对称加密，所谓对称加密就是，将信息和私钥通过某种算法混合在一起，这样除非知道私钥，不然无法获取内容，而正好客户端和服务端都知道这个私钥，所以只要加密算法够彪悍，私钥够复杂，数据就够安全。

7、传输加密后的信息

这部分信息是服务段用私钥加密后的信息，可以在客户端被还原。

8、客户端解密信息

客户端用之前生成的私钥解密服务段传过来的信息，于是获取了解密后的内容，整个过程第三方即使监听到了数据，也束手无策。

## http1/http2

http1 的缺点

高延迟-队头阻塞

无状态特性

明文传输

不支持服务端推送

http2 的特点

二进制传输

header压缩

多路复用

服务端推送

http2的缺点

tcp链接的队头阻塞并没有被完全解决（丢包情况下，会导致所有链接等待）

## tcp与udp

|  | **UDP** | **TCP** |
| --- | --- | --- |
| 是否连接 | 无连接 | 面向连接 |
| 是否可靠 | 不可靠传输，不使用流量控制和拥塞控制 | 可靠传输，使用流量控制和拥塞控制 |
| 连接对象个数 | 支持一对一，一对多，多对一和多对多交互通信 | 只能是一对一通信 |
| 传输方式 | 面向报文 | 面向字节流 |
| 首部开销 | 首部开销小，仅8字节 | 首部最小20字节，最大60字节 |
| 适用场景 | 适用于实时应用（IP电话、视频会议、直播等） | 适用于要求可靠传输的应用，例如文件传输 |

## tcp链接的过程

三次握手，四次挥手

## 请求头常用字段

1. Accept
   1. Accept: text/html  浏览器可以接受服务器回发的类型为 text/html。
   2. Accept: \*/\*   代表浏览器可以处理所有类型,(一般浏览器发给服务器都是发这个)。
2. Accept-Encoding
   1. Accept-Encoding: gzip, deflate 浏览器申明自己接收的编码方法，通常指定压缩方法，是否支持压缩，支持什么压缩方法（gzip，deflate），（注意：这不是只字符编码）。
3. Accept-Language

Accept-Language:zh-CN,zh;q=0.9  浏览器申明自己接收的语言。

1. Connection
   1. Connection: keep-alive  当一个网页打开完成后，客户端和服务器之间用于传输HTTP数据的TCP连接不会关闭，如果客户端再次访问这个服务器上的网页，会继续使用这一条已经建立的连接。
   2. Connection: close 代表一个Request完成后，客户端和服务器之间用于传输HTTP数据的TCP连接会关闭， 当客户端再次发送Request，需要重新建立TCP连接。
2. Host（发送请求时，该报头域是必需的）
   1. Host:www.baidu.com 请求报头域主要用于指定被请求资源的Internet主机和端口号，它通常从HTTP URL中提取出来的。
3. Referer
   1. Referer:https://www.baidu.com/?tn=62095104\_8\_oem\_dg 当浏览器向web服务器发送请求的时候，一般会带上Referer，告诉服务器我是从哪个页面链接过来的，服务器籍此可以获得一些信息用于处理。
4. User-Agent
   1. User-Agent:Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36 告诉HTTP服务器， 客户端使用的操作系统和浏览器的名称和版本。
5. Cache-Control
   1. Cache-Control:private 默认为private  响应只能够作为私有的缓存，不能再用户间共享
   2. Cache-Control:public 响应会被缓存，并且在多用户间共享。正常情况, 如果要求HTTP认证,响应会自动设置为 private.
   3. Cache-Control:must-revalidate  响应在特定条件下会被重用，以满足接下来的请求，但是它必须到服务器端去验证它是不是仍然是最新的。
   4. Cache-Control:no-cache  响应不会被缓存,而是实时向服务器端请求资源。
   5. Cache-Control:max-age=10 设置缓存最大的有效时间，但是这个参数定义的是时间大小（比如：60）而不是确定的时间点。单位是[秒 seconds]。
   6. Cache-Control:no-store 在任何条件下，响应都不会被缓存，并且不会被写入到客户端的磁盘里，这也是基于安全考虑的某些敏感的响应才会使用这个。
6. Cookie

Cookie是用来存储一些用户信息以便让服务器辨别用户身份的（大多数需要登录的网站上面会比较常见），比如cookie会存储一些用户的用户名和密码，当用户登录后就会在客户端产生一个cookie来存储相关信息，这样浏览器通过读取cookie的信息去服务器上验证并通过后会判定你是合法用户，从而允许查看相应网页。当然cookie里面的数据不仅仅是上述范围，还有很多信息可以存储是cookie里面，比如sessionid等。

1. Range（用于断点续传）

Range:bytes=0-5 指定第一个字节的位置和最后一个字节的位置。用于告诉服务器自己想取对象的哪部分。

**常用的http响应头**

1. Cache-Control（对应请求中的Cache-Control）
   1. Cache-Control:private 默认为private 响应只能够作为私有的缓存，不能再用户间共享
   2. Cache-Control:public 浏览器和缓存服务器都可以缓存页面信息。
   3. Cache-Control:must-revalidate 对于客户机的每次请求，代理服务器必须想服务器验证缓存是否过时。
   4. Cache-Control:no-cache 浏览器和缓存服务器都不应该缓存页面信息。
   5. Cache-Control:max-age=10  是通知浏览器10秒之内不要烦我，自己从缓冲区中刷新。
   6. Cache-Control:no-store 请求和响应的信息都不应该被存储在对方的磁盘系统中。
2. Content-Type
   1. Content-Type：text/html;charset=UTF-8 告诉客户端，资源文件的类型，还有字符编码，客户端通过utf-8对资源进行解码，然后对资源进行html解析。通常我们会看到有些网站是乱码的，往往就是服务器端没有返回正确的编码。
3. Content-Encoding
   1. Content-Encoding:gzip 告诉客户端，服务端发送的资源是采用gzip编码的，客户端看到这个信息后，应该采用gzip对资源进行解码。
4. Date
   1. Date: Tue, 03 Apr 2018 03:52:28 GMT 这个是服务端发送资源时的服务器时间，GMT是格林尼治所在地的标准时间。http协议中发送的时间都是GMT的，这主要是解决在互联网上，不同时区在相互请求资源的时候，时间混乱问题。
5. ServerServer：Tengine/1.4.6  这个是服务器和相对应的版本，只是告诉客户端服务器信息。
6. Transfer-Encoding
   1. Transfer-Encoding：chunked 这个响应头告诉客户端，服务器发送的资源的方式是分块发送的。一般分块发送的资源都是服务器动态生成的，在发送时还不知道发送资源的大小，所以采用分块发送，每一块都是独立的，独立的块都能标示自己的长度，最后一块是0长度的，当客户端读到这个0长度的块时，就可以确定资源已经传输完了。
7. Expires
   1. Expires:Sun, 1 Jan 2000 01:00:00 GMT 这个响应头也是跟缓存有关的，告诉客户端在这个时间前，可以直接访问缓存副本，很显然这个值会存在问题，因为客户端和服务器的时间不一定会都是相同的，如果时间不同就会导致问题。所以这个响应头是没有Cache-Control：max-age=\*这个响应头准确的，因为max-age=date中的date是个相对时间，不仅更好理解，也更准确。
8. Last-Modified
   1. Last-Modified: Dec, 26 Dec 2015 17:30:00 GMT 所请求的对象的最后修改日期(按照 RFC 7231 中定义的“超文本传输协议日期”格式来表示)
9. Connection
   1. Connection：keep-alive 这个字段作为回应客户端的Connection：keep-alive，告诉客户端服务器的tcp连接也是一个长连接，客户端可以继续使用这个tcp连接发送http请求。
10. Etag
    1. ETag: "737060cd8c284d8af7ad3082f209582d" 就是一个对象（比如URL）的标志值，就一个对象而言，比如一个html文件，如果被修改了，其Etag也会别修改，所以，ETag的作用跟Last-Modified的作用差不多，主要供WEB服务器判断一个对象是否改变了。比如前一次请求某个html文件时，获得了其 ETag，当这次又请求这个文件时，浏览器就会把先前获得ETag值发送给WEB服务器，然后WEB服务器会把这个ETag跟该文件的当前ETag进行对比，然后就知道这个文件有没有改变了。
11. Refresh
    1. Refresh: 5; url=http://baidu.com  用于重定向，或者当一个新的资源被创建时。默认会在5秒后刷新重定向。
12. Access-Control-Allow-Origin
    1. Access-Control-Allow-Origin: \*   \*号代表所有网站可以跨域资源共享，如果当前字段为\*那么
    2. Access-Control-Allow-Credentials就不能为trueAccess-Control-Allow-Origin: www.baidu.com 指定哪些网站可以跨域资源共享
13. Access-Control-Allow-Methods
    1. Access-Control-Allow-Methods：GET,POST,PUT,DELETE  允许哪些方法来访问
14. Access-Control-Allow-Credentials
    1. Access-Control-Allow-Credentials: true  是否允许发送cookie。默认情况下，Cookie不包括在CORS请求之中。设为true，即表示服务器明确许可，Cookie可以包含在请求中，一起发给服务器。这个值也只能设为true，如果服务器不要浏览器发送Cookie，删除该字段即可。如果access-control-allow-origin为\*，当前字段就不能为true

15.Content-RangeContent-Range: bytes 0-5/7877 指定整个实体中的一部分的插入位置，他也指示了整个实体的长度。在服务器向客户返回一个部分响应，它必须描述响应覆盖的范围和整个实体长度。

## 参考

[面试汇总-链接](https://juejin.cn/post/7016593221815910408" \l "heading-7)

[http1/http2/http3](https://juejin.cn/post/6995109407545622542)

# 浏览器相关

## 从输入URL到页面加载的全过程

## 跨域问题

**为什么会出现跨域？**