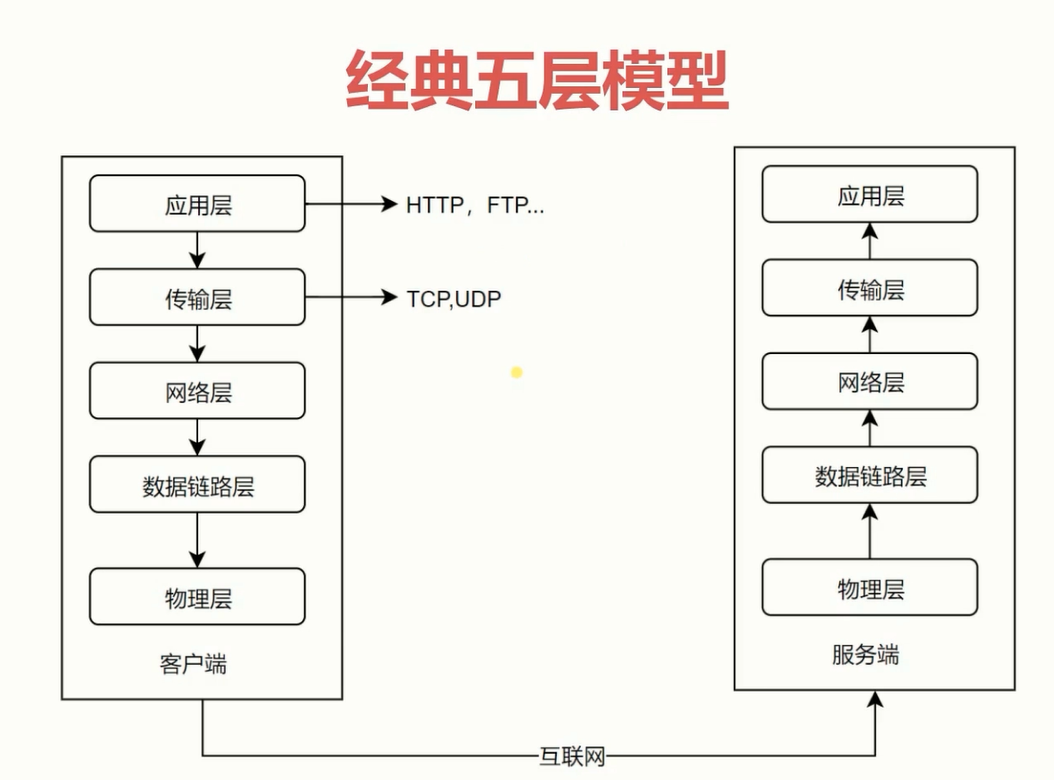
# 简介http协议

主要介绍http发展历史和浏览器报文



物理层：主要作用是定义物理设备如何传输数据。如网卡，光缆

数据链路层：在通信的实体间建立数据链路连接。如软件

网络层：为数据在结点之间传输创建逻辑链路。如我们的电脑是如何连接服务器的

传输层：向用户提供可靠的端到端服务，传输层向高层屏蔽了下层数据通信细节

应用层：为应用软件提供了很多服务，构建与TCP协议之上，屏蔽网络传输相关的细节

发展历史：

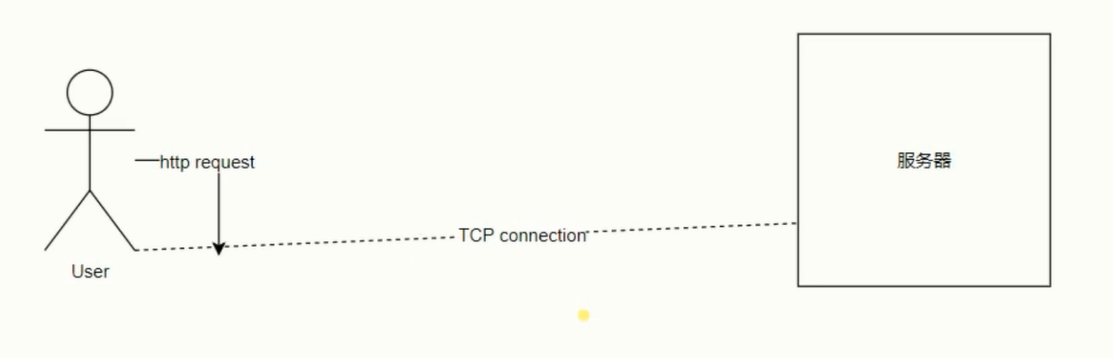
HTTP/0.9：只有一个命令get ,没有HEADER等描述数据的信息，服务器发送完毕，就关闭TCP连接

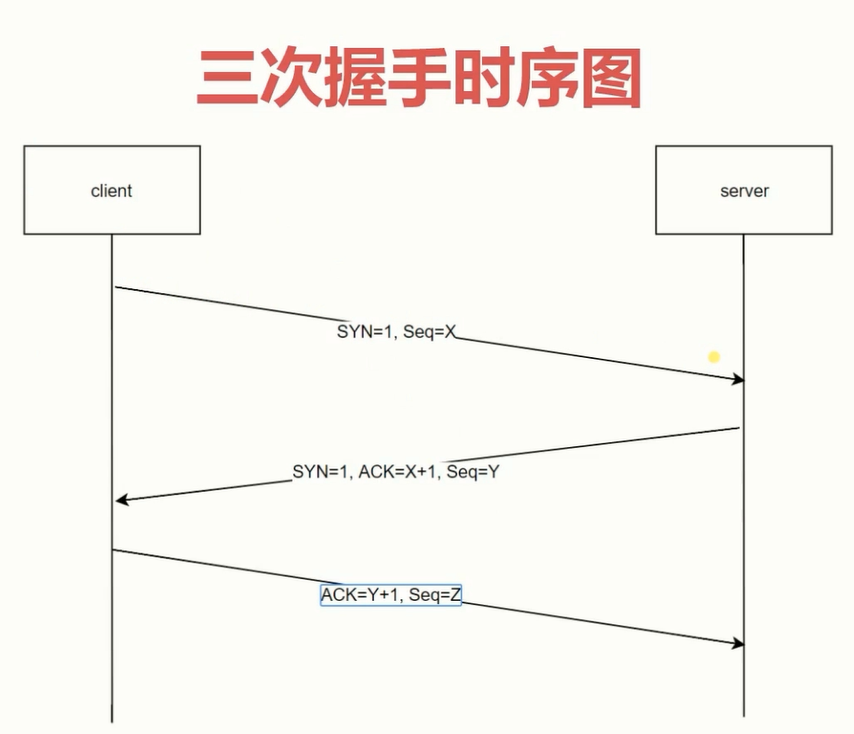
HTTP/1.0：增加了很多命令（如，post,delete,patch等）;增加了status，code和header,多字符集的支持，多部分的发送，权限，缓存等（浏览器报文信息）

HTTP/1.1：支持[持久连接](https://zhuanlan.zhihu.com/p/57142432)，同一个连接可以发送多个请求，增加了host（可以在同一台服务器跑多个服务）和其他一些命令

HTTP2：所有数据以二进制传输，同一链接发送多个请求可以不再需要按照顺序来（[信道复用](https://www.jianshu.com/p/7612431a3eda)），头信息压缩以及推送（服务端可以主动发信息，当请求html时可以主动发送，js和css文件，加快传输效率）等

创建连接要经过3次握手





RUI包含URL和URN:Uniform Rerource Identifier/统一资源标志符：识别定位资源的位置

URL:Uniform Rerource Locator/统一资源定位器

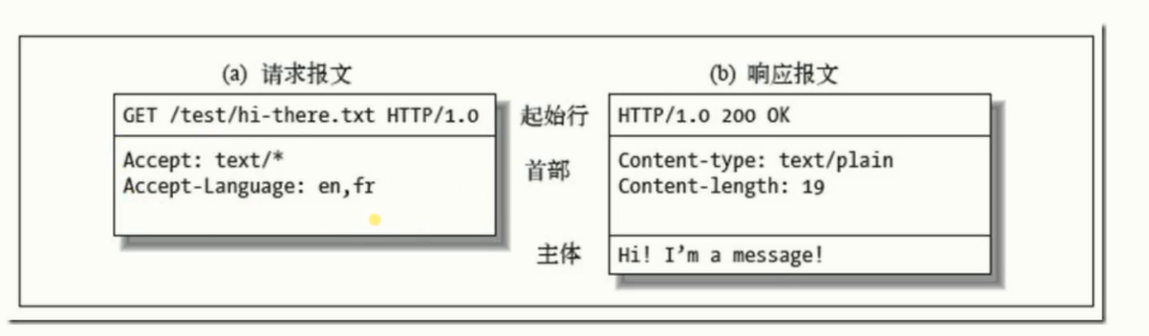
(ftp) http(s)://user:pass@host.com:80/path?query=string#hash

host.com域名 通过dns 解析成 ip ，找到相应地址

#hash进行锚点定位

URN:永久统一资源定位符（当资源搬离原来的服务器，还能找到原来的资源，目前还没有成熟的解决方案）

报文：



100~199 请求要持续进行 200~299 成功 300~399 请求需要重定向 400~499 请求又问题

500~599 服务器故障

Request Headers :

POST /user/signin HTTP/1.1    --请求方式 文件名 http版本号   
Host: passport.cnblogs.com     --请求地址  
Connection: keep-alive

--Connection决定当前的事务完成后，是否会关闭网络连接。如果该值是“keep-alive”，网络连接就是持久的，不会关闭，使得对同一个服务器的请求可以继续在该连接上完成。  
Content-Length: 557             --发送给HTTP服务器的长度  
Origin: https://passport.cnblogs.com        --起源是来自哪里  
X-Requested-With: XMLHttpRequest       --表明是ajax异步请求  
Referer:https://passport.cnblogs.com/user/signin?ReturnUrl=http://www.cnblogs.com/fighter007/p/8422868.html --提供上下文服务器，告诉服务器我是从哪里来的，一般用于网站流量统计。  
  
Accept-Encoding: gzip, deflate, br      --浏览器申明自己接收的编码方式：通常指定压缩、是否支持压缩、支持什么方式压缩（gzip/default）  
Accept-Language: zh-CN,zh;q=0.9     --浏览器申明自己接收的语言  
Accept: application/json, text/javascript, \*/\*; q=0.01--浏览器接收的媒体类型 application/json, text/javascript  \*/\* 代表浏览器可以处理所有类型  
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36  --告诉HTTP服务器客户端浏览器使用的操作系统和浏览器的版本和名称

Content-Type: application/json; charset=UTF-8     --浏览器接收的内容类型、字符集

补充：常见的媒体格式Content-Type类型如下：

    text/html ： HTML格式

    text/plain ：纯文本格式

    text/xml ：  XML格式

    image/gif ：gif图片格式

    image/jpeg ：jpg图片格式

    image/png：png图片格式

   以application开头的媒体格式类型：

   application/xhtml+xml ：XHTML格式

   application/xml     ： XML数据格式

   application/atom+xml  ：Atom XML聚合格式

   application/json    ： JSON数据格式

   application/pdf       ：pdf格式

   application/msword  ： Word文档格式

   application/octet-stream ： 二进制流数据（如常见的文件下载）

   application/x-www-form-urlencoded ： <form encType=””>中默认的encType，form表单数据被编码为key/value格式发送到服务器（表单默认的提交数据的格式）

另外一种常见的媒体格式是上传文件之时使用的：

 multipart/form-data ： 需要在表单中进行文件上传时，就需要使用该格式

Response Headers

HTTP/1.1 200 OK        --响应的状态码 200表示正常应答  
Date: Mon, 12 Feb 2018 11:22:13 GMT              --生成消息的具体时间和日期  
Content-Type: application/json; charset=utf-8   --http服务器告诉浏览器自己响应的对象类型和字符集（并且告诉客户端实际返回的内容的内容类型）

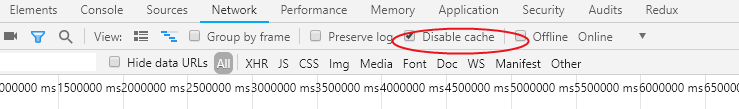
更多类型，例如：  
Content-Type:text/html; charset=utf-8  
Content-Type:text/html;charset=GB2312  
Content-Type: image/jpeg

Content-Length: 54               --http服务器的响应实体正文的长度  
Content-Language：da        --http服务器告诉浏览器自己响应的语言  
Content-Encoding：gzip      --http服务器表名自己使用了什么压缩方法  
Connection: keep-alive         --Connection决定当前的事务完成后，是否会关闭网络连接。如果该值是“keep-alive”，网络连接就是持久的，不会关闭，使得对同一个服务器的请求可以继续在该连接上完成   
Cache-Control:

-- "private" 表示该响应是专用于某单个用户的，中间人不能缓存此响应，该响应只能在请求的浏览器中进行缓存。

--"public" 指令表示该响应可以被任何中间人。（浏览器，代理服务器）  
若指定了"public"，则一些通常不被中间人缓存的页面（因为默认是private）  
（比如 带有HTTP验证信息（帐号密码）的页面 或 某些特定影响状态码的页面），将会被其缓存。  
--no-cache 任何节点都不可以缓存（包括服务器）

浏览器控制台勾选了Disable cache 浏览器不缓存

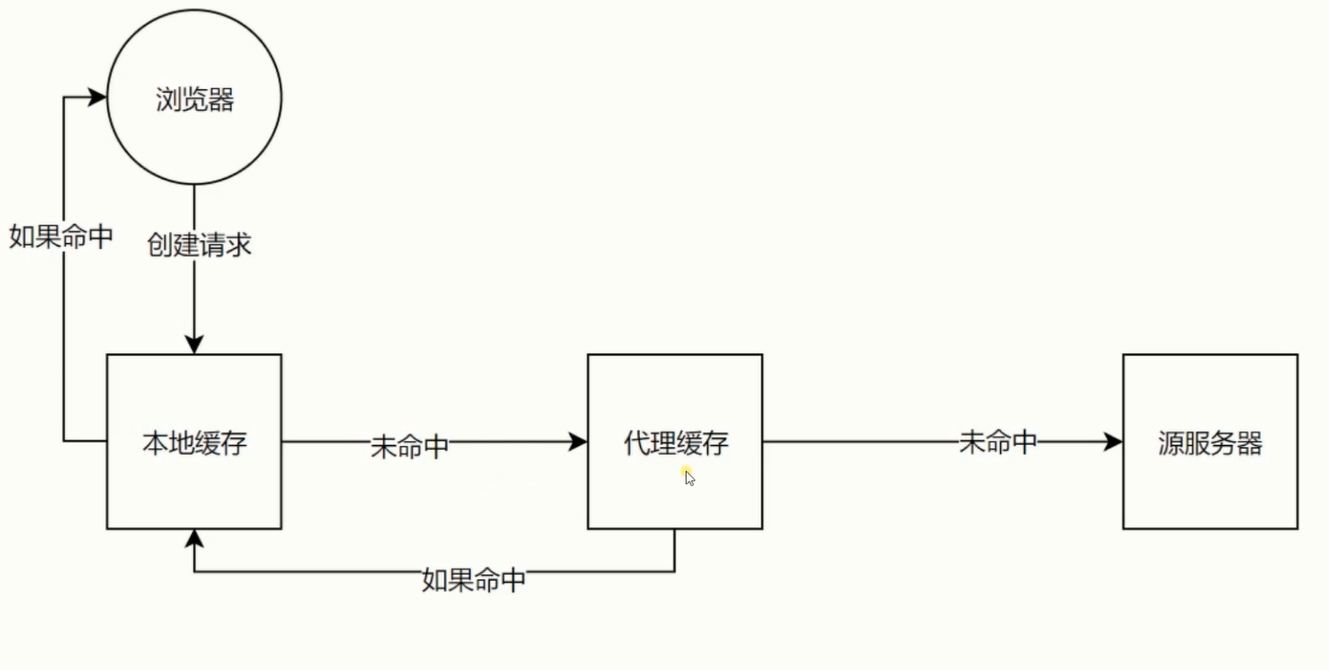


--max-age =<second> 缓存过期时间 单位s

--max-age =“<second>，no-cache”,配合Last-Modified:(Wed, 21 Dec 2011 09:09:10 GMT        --申明资源的最后修改日期和时间)和tag:（当浏览器请求服务器的某项资源(A)时, 服务器根据A算出一个哈希值(3f80f-1b6-3e1cb03b)并通过 ETag 返回给浏览器）进行缓存验证

--s-maxage :代理服务器缓存过期时间

--max-state：在请求头设置，客户端主动设置缓存过期时间

  
\*\*\*\*\*

X-AspNetMvc-Version: 5.2         --asp.net版本号  
X-AspNet-Version: 4.0.30319     --asp.net技术的版本号  
X-Powered-By: ASP.NET            --表示网站是由什么技术开发  
Set-Cookie: key=value是非常重要的header, 用于把cookie 发送到客户端浏览器,每一个写入cookie都会生成一个Set-Cookie,可以同时设置多个[key=value,key2=value2],可以设置过期时间”key=value;max-age=x”(x单位为秒,不设置浏览器关闭时，自动清除)，还可以加”key=value;httpOnly”只可以设置cookie,禁止js访问该cookie,”key=value;domain=test.com”--设置后，a.test.com域名下也可以访问该cookie

Redriect:

Location:’新路径’ //如果返回码为302每次请求旧路径都会重定向到新路径，如果返回码为301，第二次及以后请求旧路径，都不会先到旧路径再重定向到新路径，而是直接到新路径，除非客户端清楚缓存，所以返回码301慎用。