# 参考

https://www.jianshu.com/p/5964697973a8

# http 协议

## 基础

* Hyper text transfer protocol

· protocol：首先这是一个协议，是个约定，确定计算机交流的通信规范以及各种控制和错误处理方式。

· transfer：这是一个传输协议，其基于TCP或者UDP，将信息从一个地方传送到另外一个地方。并且传送过程中允许代理。

· hyper text：超文本，传送包括文本，音频，视频等。

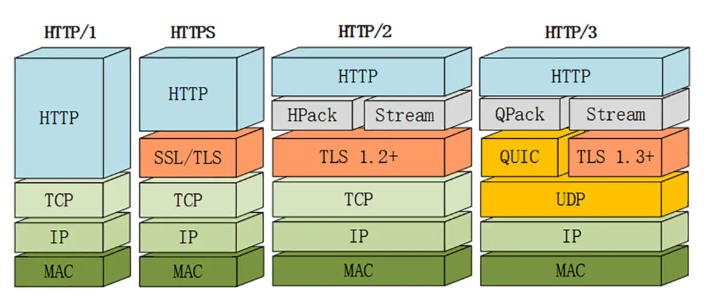
## 介绍

HTTP协议处于四层网络协议模型的应用层，是互联网世界的基础。

在HTTP协议上，有衍生封装了很多rpc应用，例如说SOAP调用。

## 演变

HTTP并不是一个静止的协议，而是一个不断演化的协议，正式因为生命力强，所以才不断演化



|  |
| --- |
| · HTTP从一开始比较简单的原型，到各种优化，例如说长连接、cookie等优化，最终写入rfc，标志了HTTP被世界认可，真正广泛应用。这是HTTP/1  · 为了解决安全问题，在tcp和http层之间增加了一个中间层，SSL/TLS，解决加密解密的安全问题。这是HTTPS  · 为了解决HTTP/1的排队阻塞问题，以及安全等问题，google利用自身的巨大优势，将自身的方案提案为HTTP/2  · 还是google，提出并利用UDP作为传输层协议，这就是HTTP/3 |

## 特点

· HTTP是可扩展的。不同于SNMP等协议，HTTP的协议头除了公共约束，是可以扩展的，所以后面才产生若干优化。

· HTTP是可靠的，正如上面HTTP演化所述，HTTP/1、HTTPS、HTTP/2都是基于TCP的，而HTTP/3虽然是基于UDP，但是实际上是在UDP上层，把TCP的控制等专门提取出一个协议，然后在这个协议上开发HTTP/3，所以说HTTP/3也是可靠的。

· HTTP可以传输任意内容，如上所述，可以封装其它协议，如果SOAP，这是SOA架构的基础。

· HTTP是无状态的，每个请求都毫无关联，为了实现会话等，所以有cookie。

## 方法

· GET 获取读取资源

· HEAD 获取资源的源信息

· POST向资源提交数据，写入上传数据

· PUT向资源提交数据，返回无body

· DELETE删除

· CONNECT 建立特殊连接隧道

· OPTIONS 列出对资源实行的方法

· TRACE 列出请求响应的传输路径

|  |
| --- |
| · 例如现在网站都基本要求用https访问，而用户输入http，服务器可以返回301，并指向https的URL。  · 负载均衡。用户向负载均衡服务器发送请求，负载均衡服务器计算出一个请求服务器地址，然后给用户返回301和新的地址，用户接着访问新的地址。这种方式不太常用。 |

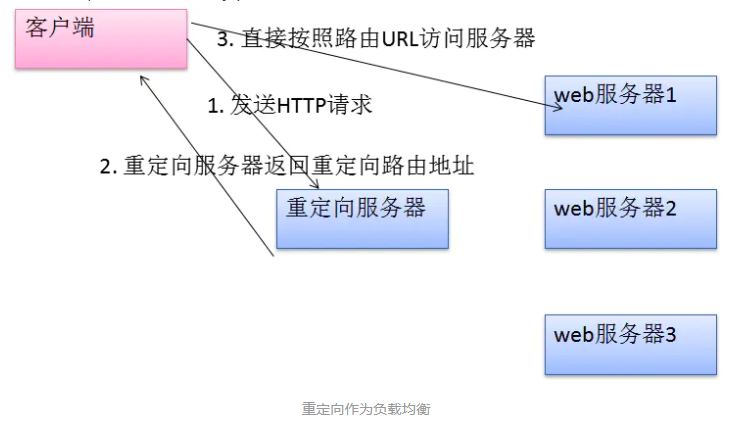
## 状态码

## DNS

## 重定向

|  |
| --- |
| * 浏览器首先发送一条HTTP请求 * 服务器返回状态码301、302 * 浏览器检查到状态码301、302，就再次发送HTTP请求，该请求的URL在返回报文的location字段中。   这个过程，并不是HTTP协议本身进行了跳转，而是浏览器根据HTTP协议定义的状态码和扩展出的头字段location进行相应的跳转操作。 |

* 使用场景



|  |
| --- |
| 访问过程如上 ：  · 访问过程如上，中间加上了重定向服务器作为负载均衡。这种方式由于要损耗性能，实际上很少用。  · 永久跳转。当某个网站地址已经失效，此时可以使用永久跳转，直接导入到新网址，直到不再有人使用旧网址，就可以将旧网址失效。  · 临时跳转。当前网址临时有问题或者处于运维考虑，临时跳转到另外一个新地址，当运维结束，就不再跳转。 |

## HTTP cokies

## HTTP缓存控制

* HTTP本身没有缓存功能，所以只是在HTTP协议中扩充了缓存字段，让浏览器和服务端按照缓存控制字段处理是否需要缓存
* 服务器端使用Cache-Control, 例如Cache-Control: max-age=30，就是说服务器端发出报文到客户端智能保留30秒。浏览器是怎么处理的呢，浏览器收到缓存条件就把数据缓存起来，如果没有过期就可以直接使用，过期就要去服务器验证是否仍然可用。
* 浏览器验证

· 在第一次缓存的时候浏览器收到的服务器报文中，有Last-Modified和ETag两个字段。

· 浏览器接下来的处理中，会先下发HEAD请求，报文带上if-Modified-Since和if-None-Match两个字段以及上面Last-Modified和ETag两个字段, 交给服务器验证，服务器验证后如果返回304 Not Modified，就说明缓存依然有效，浏览器就可以继续使用缓存

## 代理

* 代理是HTTP比较常见的应用，最常用的代理应用是负载均衡

· 代理服务器可以对服务器做管理，建立心跳，如果服务器已经异常，则端口服务器连接，保证服务高可用性。

· 黑名单，可以在代理服务器上设置黑名单，保证服务器高可用。

· 缓存，如ngix，可以用来作为缓存静态数据，如果发现是静态数据，就直接从代理服务器返回。

|  |
| --- |
| 由于中间存在代理，服务器和客户端之间就不能互相知道IP，而服务器的IP一般按照安全原则是不暴漏给客户端的，但是服务器确有必要知道客户端的IP地址，这时有两个字段 X-Forwarded-Host和X-Real-IP只记录客户端的信息。 |

# https

## 