

常用的HTTP方法有哪些？

GET方法与POST方法的区别

区别一

区别二

区别三

区别四

区别五

HTTP请求报文与响应报文格式

请求报文包含三部分

响应报文包含三部分

常见的HTTP相应状态码

常见HTTP首部字段

通用首部字段（请求报文与响应报文都会使用的首部字段）

请求首部字段（请求报文会使用的首部字段）

响应首部字段（响应报文会使用的首部字段）

实体首部字段（请求报文与响应报文的的实体部分使用的首部字段）

HTTP的缺点与HTTPS

HTTP优化

HTTP1.0和HTTP1.1的一些区别

长链接

节约带宽

HOST域

HTTP 1.1、HTTP2.0主要区别

多路复用

数据压缩

服务器推送

常用的HTTP方法有哪些？

GET： 用于请求访问已经被URI（统一资源标识符）识别的资源，可以通过URL传参给服务器。

POST： 用于传输信息给服务器，主要功能与GET方法类似，但一般推荐使用POST方式。

PUT： 传输文件，报文主体中包含文件内容，保存到对应URI位置。

HEAD： 获得报文首部，与GET方法类似，只是不返回报文主体，一般用于验证URI是否有效。

DELETE： 删除文件，与PUT方法相反，删除对应URI位置的文件。

OPTIONS： 查询相应URI支持的HTTP方法

GET方法与POST方法的区别

区别一

get重点在从服务器上获取资源，post重点在向服务器发送数据；

区别二

get传输数据是通过URL请求，以field（字段）= value的形式，置于URL后，并用"?"连接，多个请求数据间用"&"连接，如

http://127.0.0.1/Test/login.action?name=admin&password=admin，这

这个过程用户是可见的；

post传输数据通过Http的post机制，将字段与对应值封存在请求实体中发送给服务器，这个过程对用户是不可见的；

区别三

Get传输的数据量小，因为受URL长度限制，但效率较高；

Post可以传输大量数据，所以上传文件时只能用Post方式；

区别四

get是不安全的，因为URL是可见的，可能会泄露私密信息，如密码等；

post较get安全性较高；

区别五

get方式只能支持ASCII字符，向服务器传的中文字符可能会乱码。

post支持标准字符集，可以正确传递中文字符

HTTP请求报文与响应报文格式

请求报文包含三部分

- a、请求行：包含请求方法、URI、HTTP版本信息
- b、请求首部字段
- c、请求内容实体

响应报文包含三部分

- a、状态行：包含HTTP版本、状态码、状态码的原因短语
- b、响应首部字段
- c、响应内容实体

常见的HTTP相应状态码

1xx：指示信息--表示请求已接收，继续处理

2xx：成功--表示请求已被成功接收、理解、接受

3xx：重定向--要完成请求必须进行更进一步的操作

4xx：客户端错误--请求有语法错误或请求无法实现

5xx: 服务器端错误--服务器未能实现合法的请求

常见HTTP首部字段

通用首部字段（请求报文与响应报文都会使用的首部字段）

Date: 创建报文时间

Connection: 连接的管理

Cache-Control: 缓存的控制

Transfer-Encoding: 报文主体的传输编码方式

请求首部字段（请求报文会使用的首部字段）

Host: 请求资源所在服务器

Accept: 可处理的媒体类型

Accept-Charset: 可接收的字符集

Accept-Encoding: 可接受的内容编码

Accept-Language: 可接受的自然语言

响应首部字段（响应报文会使用的首部字段）

Accept-Ranges: 可接受的字节范围

Location: 令客户端重新定向到的URI

Server: HTTP服务器的安装信息

实体首部字段（请求报文与响应报文的实体部分使用的首部字段）

Allow: 资源可支持的HTTP方法

Content-Type: 实体主类的类型

Content-Encoding: 实体主体适用的编码方式

Content-Language: 实体主体的自然语言

Content-Length: 实体主体的字节数

Content-Range: 实体主体的位置范围，一般用于发出部分请求时使用

HTTP的缺点与HTTPS

- a、通信使用明文不加密，内容可能被窃听
- b、不验证通信方身份，可能遭到伪装
- c、无法验证报文完整性，可能被篡改

HTTPS就是HTTP加上加密处理（一般是SSL安全通信线路）+认证+完整性保护

HTTP优化

利用负载均衡优化和加速HTTP应用

利用HTTP Cache来优化网站

HTTP1.0和HTTP1.1的一些区别

长链接

HTTP 1.0需要使用keep-alive参数来告知服务器端要建立一个长连接，而HTTP1.1默认支持长连接。

HTTP是基于TCP/IP协议的，创建一个TCP连接是需要经过三次握手的,有一定的开销，如果每次通讯都要重新建立连接的话，对性能有影响。因此最好能维持一个长连接，可以用个长连接来发多个请求。

节约带宽

HTTP 1.1支持只发送header信息(不带任何body信息)，如果服务器认为客户端有权限请求服务器，则返回100，否则返回401。客户端如果接受到100，才开始把请求body发送到服务器。

这样当服务器返回401的时候，客户端就可以不用发送请求body了，节约了带宽。

另外HTTP还支持传送内容的一部分。这样当客户端已经有一部分的资源后，只需要跟服务器请求另外的部分资源即可。这是支持文件断点续传的基础

HOST域

现在可以web server例如tomcat，设置虚拟站点是非常常见的，也即是说，web server上的多个虚拟站点可以共享同一个ip和端口。

HTTP1.0是没有host域的，HTTP1.1才支持这个参数

HTTP 1.1、HTTP2.0主要区别

多路复用

HTTP2.0使用了多路复用的技术，做到同一个连接并发处理多个请求，而且并发请求的数量比HTTP1.1大了好几个数量级。

当然HTTP1.1也可以多建立几个TCP连接，来支持处理更多并发的请求，但是创建TCP连接本身也是有开销的。

TCP连接有一个预热和保护的过程，先检查数据是否传送成功，一旦成功过，则慢慢加大传输速度。因此对应瞬时并发的连接，服务器的响应就会变慢。所以最好能使用一个建立好的连接，并且这个连接可以支持瞬时并发的请求。

数据压缩

HTTP1.1不支持header数据的压缩，HTTP2.0使用HPACK算法对header的数据进行压缩，这样数据体积小了，在网络上传输就会更快

服务器推送

当我们对支持HTTP2.0的web server请求数据的时候，服务器会顺便把一些客户端需要的资源一起推送到客户端，免得客户端再次创建连接发送请求到服务器端获取。这种方式非常合适加载静态资源。

服务器端推送的这些资源其实存在客户端的某处地方，客户端直接从本地加载这些资源就可以了，不用走网络，速度自然是快很多的