**1、简介**

HTTP 是一个属于应用层的面向对象的协议，由于其简捷、快速的方式，适用于分布式超媒体信息系统。它于1990 年提出，经过几年的使用与发展，得到不断地完善和扩展。目前在WWW 中使用的是HTTP/1.0的第六版，HTTP/1.1 的规范化工作正在进行之中，而且HTTP-NG(Next Generation of HTTP)的建议已经提出。  
HTTP 协议的主要特点可概括如下：  
1.支持客户/服务器模式。  
2.简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于HTTP 协议简单，使得HTTP 服务器的程序规模小，因而通信速度很快。  
3.灵活：HTTP 允许传输任意类型的数据对象。正在传输的类型由Content-Type 加以标记。  
4.无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。  
5.无状态：HTTP 协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快。

**2、HTTP请求与响应**

2.1 HTTP请求－－方法

|  |  |
| --- | --- |
| GET | 请求获取由Request-URI所表示的资源 |
| POST | 在Request-URI所标识的资源后附加新的数据 |
| HEAD | 请求获取由Request-URI所标识的资源的响应消息报头 |
| PUT | 请求服务器存储一个资源，并用Request-URI作为其标识 |
| DELETE | 请求服务器删除由Request-URI所标识的资源 |
| TRACE | 请求服务器回送收到的请求信息，主要用于测试或诊断 |
|  |  |
|  |  |

2.2 HTTP请求－－常见响应码

|  |  |  |
| --- | --- | --- |
| 状态码 | 描述 | 客户端请求成功 |
| 200 | OK | 客户端请求成功 |
| 206 | Partial content | 返回请求实体的部分内容 |
| 301 | Moved Permanently | 目标永久移动 |
| 302 | found | 找到目标（目标临时被移动） |
| 304 | Not Modified | 目标没有修改 |
| 400 | Bad Request | 客户端请求有语法错误，不能被服务器所理解 |
| 403 | Forbidden | 服务器收到请求，但是拒绝提供服务 |
| 404 | Not Found | 请求资源不存在 |
| 413 | Request Entity Too Large | 请求实体太大 |
| 500 | Internal Server Error | 服务器发生不可预测的错误，导致无法完成客户端的请求 |
| 502 | Bad Gateway | 网关错误，有可能是网络不通或负载过大导致。 |
| 504 | Gateway Timeout | 连接超时 |

**3、常见HTTP头域**

**[plain]** [view plaincopy](http://blog.csdn.net/xifeijian/article/details/8712455)

1. Accept       (标识浏览器能够接受的实体类型)
3. Accept-Charset /Accept-Language   (字符集和语言)
5. Accept-Encoding  (接受编码格式 gzip或者 deflat)
7. Cache-Control ( 缓存策略。例如：Cache-Control: no-cache )
9. Pragma             ( 标注。例如：Pragma: no-cache )
11. Connection       ( 连接参数。例如：Connection: close )
13. Date                  (日期。 例如：Date:Mon,31Dec200104:25:57GMT)
15. Content-Type    (标识实体类型。例如：Content-Type: text/html。  Mime.types文件例子 )
17. Transfer-Encoding  (传输编码。Chunk )
19. Content-Length       (实体长度)
21. Set-Cookie / Cookie
23. User-Agent       (浏览器标识)
25. Host                  (目标主机)
27. Expires
29. Last-Modified
31. Range/Content-Range
33. Content-Disposition
35. Referer              (引用，通常是上一个访问的URL)

**4、HTTP协议的格式和结构---例子**

**[plain]** [view plaincopy](http://blog.csdn.net/xifeijian/article/details/8712455)

1. GET /index.shtml HTTP/1.1
2. Host: www.google.com
3. User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; zh-CN; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5
4. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8
5. Accept-Language: zh-cn,zh;q=0.5
6. Accept-Encoding: gzip,deflate
7. Accept-Charset: gb2312,utf-8;q=0.7,\*;q=0.7
8. Keep-Alive: 300
9. Connection: keep-alive
11. HTTP/1.x 200 OK
12. Date: Tue, 10 Feb 2009 08:26:37 GMT
13. Server: Apache/2.2.8 (Unix) PHP/5.2.6
14. Accept-Ranges: bytes
15. Keep-Alive: timeout=5, max=99
16. Connection: Keep-Alive
17. Transfer-Encoding: chunked
18. Content-Type: text/html; Charset=utf-8