<http://user:passwd@www.example.jp:80/dir/index.htm?uid=1#ch1>

http:// -- 协议方案名称

Uer:passwd -- 用户名和密码

[www.example.jp](http://www.example.jp) -- 服务器地址，使用绝对的URI必须制定待访问的服务器地址，地址可以是类似hackr.jp这种DNS可解析的名称，或是192.168.1.1这类IPv4地址名，还可以是[0:0:0:0:0:0:0:1]这样用方括号括起来的IPv6地址名。

/dir/index.htm -- 带层次的文件路径

Uid=1 -- 查询字符串，针对已制定的文件路径内的资源，可以使用查询字符串传入任意参数。此项可选。

Ch1 -- 片段识别符，使用片段标识符通常可标记出以获取资源中的子资源。

## 简单的HTTP协议

HTTP协议规定，请求从客户端发出，最后服务器端响应该请求并返回。

GET /index.htm HTTP/1.1

Host: hackr.jp

起始行GET表示请求访问服务器的类型，称为方法。

随后的/index.htm指明了请求访问的资源对象，一叫做请求URI

最后的HTTP/1.1表示HTTP的版本号，用来提示客户端使用的HTTP协议功能。

含义：请求访问某台HTTP服务器上的/index.htm页面资源。

**请求报文是有请求方法、请求URI、协议版本、可选的请求首部字段和内容实体构成的。**

HTTP/1.1虽然是无状态协议，但为了实现期望的保持状态功能，于是引入了Cookie技术。有了Cookie技术，就可以管理状态了。

请求URI的方式有很多种，例：

·URI为完整的请求URI

GET <http://hackr.jp/index.htm> HTTP/1.1

·在首部字段Host中写明网络域名或IP地址

GET /index.htm HTTP/1.1

Host: hackr.jp

·如果不是访问特定资源而是对服务器本身发起请求，可以

用一个\*来代替请求URI。

OPTIONS \* HTTP/1.1

### 2.5 告知服务器意图的HTTP方法

**GET：获取资源**

GET方法用来请求访问已被URI识别的资源。如果请求的资源是文本，那就保持原样返回，如果是CGI(Common Gateway Interface)那样的程序，返回经过执行后的输出结果。

|  |  |
| --- | --- |
| 请求 | GET /index.htm HTTP/1.1  Host:www.hack.jp |
| 响应 | 返回index.html的页面资源 |

|  |  |
| --- | --- |
| 请求 | GET /index.htm HTTP/1.1  Host:www.hack.jp  If-Modified-Since: Thu,12 Jul 2012 07:30:00 GMT |
| 响应 | 仅返回2012年7月12日7点30分以后更新过的index.html页面资源，如果没有内容更新，则以状态码304 Not Modified作为响应返回 |

**POST：传输实体主体**

POST方法用来传输实体的主体。

|  |  |
| --- | --- |
| 请求 | POST /submit.cgi HTTP/1.1  Host:www.hack.jp  Content-Length:1560 （1560字节的数据） |
| 响应 | 返回submit.cgi接收数据的处理结果 |

**PUT：传输文件**

PUT方法用来传输文件，但HTTP/1.1的PUT方法自身不带验证机制，任何人都可以上传文件，存在安全问题，因此一般的web网站不使用该功能。

|  |  |
| --- | --- |
| 请求 | PUT /example.html HTTP/1.1  Host:www.hack.jp  Content-Type: text/html  Content-Length: 1560 （1560字节的数据） |
| 响应 | 返回状态码204 No Content |

**HEAD：获得报文首部**

HEAD方法和GET方法一样，只是不返回报文的主体部分，用于确认URI的有效性及资源更新的日期时间等。

|  |  |
| --- | --- |
| 请求 | HEAD /index.html HTTP/1.1  Host:www.hack.jp |
| 响应 | 返回index.html有关响应的首部 |

**DELETE：删除文件**

DELETE与PUT方法一样，HTTP/1.1协议都不带验证机制，所以一般的web网站也不适用DELETE方法。

|  |  |
| --- | --- |
| 请求 | DELETE /submit.cgi HTTP/1.1  Host:www.hack.jp |
| 响应 | 返回状态码204 No Connect |

**OPTIONS：询问支持的方法**

OPTIONS方法用于查询针对请求URI指定的资源支持的方法。

|  |  |
| --- | --- |
| 请求 | OPTIONS /submit.cgi HTTP/1.1  Host:www.hack.jp |
| 响应 | HTTP/1.1 200 OK  Allow：GET，POST，HEAD， OPTIONS |

**TRACE：追踪路径**

TRACE方法是让Web服务器端将之前的请求通信环路回给客户端的方法。

发送请求时在MAX-Forwards首部字段摊入数值，没经过一个服务器端就将该数值减一，当数值刚好减少到0时，就停止继续传输

|  |  |
| --- | --- |
| 请求 | TRACE /HTTP/1.1  Host:hackr.jp  Max-Forwards:2 |
| 响应 | HTTP/1.1 200 OK  Content-Type:message/http  Content-Length: 1024  TRACE/HTTP/1.1  Host:hackr.jp  Max-Forwards：2(返回响应包含请求内容) |

**CONNECT：要求用隧道协议连接代理**

CONNECT方法要求在与代理服务器通信时建立隧道，使用SSL（Secure Sockets Layer，安全套接层）和TLS（Transport Layer Security，传输安全层）协议把通信内容加密后经网络隧道传输。

CONNECT 代理服务器名：端口号 HTTP版本

|  |  |
| --- | --- |
| 请求 | COONECT proxy.hackr.jp:8080 HTTP/1.1  Host：proxy.hacks.jp |
| 响应 | HTTP/1.1 200 OK (之后进入网络隧道） |

### 2.7 持久连接节省通讯量

### 2.8 使用Cookie的状态管路

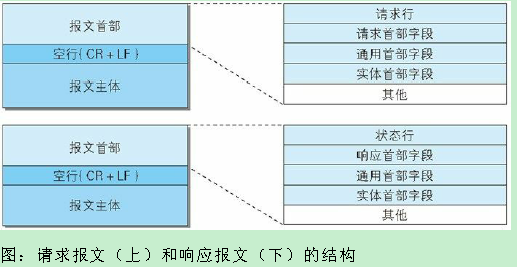
## HTTP报文内的HTTP信息

#### 3.1 HTTP报文

HTTP报文大致可分为报文首部和报文主体两块。两者由最初出现的空行（CR+LF）来划分。通常，并不一定要有报文主体。



#### 3.2 请求报文及响应报文的结构





请求行：请求方法+请求URI+HTTP版本。

状态行：响应结果的状态码+原因短语+HTTP版本。

一般有4种首部，分别是：通用首部、响应首部、请求首部和实体首部。

#### 3.3 编码提升传输速率

## 第4章 返回结果的HTTP状态码

### 4.1状态码告知从服务器端返回的请求结果

|  |  |  |
| --- | --- | --- |
|  | 类别 | 原因短语 |
| 1XX | Informational（信息性状态码） | 接收的请求正在处理 |
| 2XX | Success（成功状态码） | 请求正常处理完毕 |
| 3XX | Redirection（重定向状态码） | 需要进行附加操作以完成请求 |
| 4XX | Client Error（客户端错误状态码） | 服务器无法处理请求 |
| 5XX | Server Error（服务器错误状态码） | 服务器处理请求出错 |

### 4.2 2XX成功

|  |  |  |
| --- | --- | --- |
| 数字 | 信息 | 备注 |
| 200 | OK | 表示从客户端发来的请求在服务器端被正常处理了 |
| 204 | No Content | 响应报文中不包含实体的主体部分，一般在只需要从客户端发送信息，而对客户端不需要发送信息内容的情况下使用 |
| 206 | Partial Content | 客户端进行了范围请求，服务端成功执行了这部分的GET请求。响应报文中包含由Content-Range指定范围的试题内容。 |
| 301 | Moved Permanently | 永久性重定向。该状态码表示请求的资源已被分配了新的URI，以后应使用资源现在所指的URI。例：  当指定资源路径的最后忘记添加斜杠“/”，就会产生301状态码。http://example.com/sample |
| 302 | Found | 临时性重定向，和301相比，302代表资源不是永久移动，只是临时性质的，不需要更改保存的书签。 |
| 303 | See Other | 注1. |
| 304 | Not Modified | 客户端发送附带条件的请求时，服务器端允许请求访问资源，但未满足条件的情况。注2 |
| 307 | Temporary Redirect | 临时重定向，302标准禁止POST变换成GET，但实际使用时大家并不遵守。307会遵照浏览器的标准，不会从POST变成GET。 |
| 400 | Bad Request | 请求报文中存在语法错误。 |
| 401 | Unauthorized |  |
| 403 | Forbidden | 对请求资源的访问被服务器拒绝了。 |
| 404 | Not Found | 服务器上无法找到请求的资源。 |
| 500 | Internal Server Error | 服务器端在执行请求时发生了错误。也有可能是Web应用存在的bug或某些临时的故障 |
| 503 | Service Unavailable | 服务器暂时出于超负载或正在进行停机维护，现在无法处理请求。最好写入RetryAfter首部字段解除上述情况需要的时间。 |

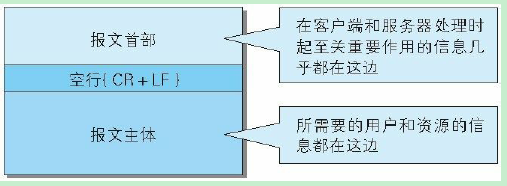
注1、303状态码表示由于请求对应的资源存在着另一个URI，应使用GET方法定向获取请求的资源。

当301、302、303响应状态码返回时，几乎所有的浏览器会把POST改成GET，并删除请求报文内的主体，之后请求会自动再次发送。

301/302标准是禁止将POST方法改变成GET方法的，但实际使用时大家都会这么做。

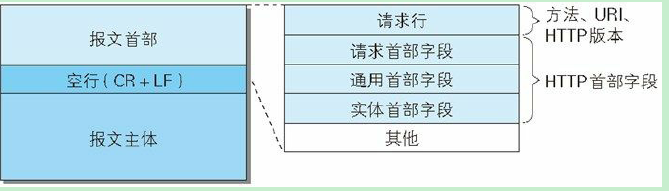
## HTTP首部

### 6.1 HTTP报文首部



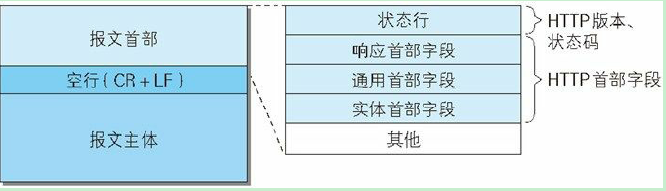
**HTTP请求报文**

HTTP报文由方法、URI、HTTP版本、HTTP首部字段等构成



**HTTP响应报文**

HTTP响应报文由HTTP版本、状态码（数字和原因短语）、HTTP首部字段3部分构成



### 6.2 HTTP首部字段

#### 6.2.1 HTTP首部字段传递重要信息

使用首部字段是为了给浏览器和服务器提供报文主体大小、所使用的语言、认证信息等内容。

#### 6.2.2 HTTP首部字段结构

首部字段名：字段值

#### 6.2.3 4中HTTP首部字段类型

通用首部字段

请求首部字段

响应首部字段

实体首部字段

#### 6.2.4 HTTP/1.1首部字段一览

表6-1：通用首部字段

|  |  |
| --- | --- |
| 首部字段名 | 说明 |
| Cache-Control | 控制缓存的行为 |
| Connection | 逐跳首部、连接的管理 |
| Date | 创建报文的日期时间 |
| Pragma | 报文指令 |
| Trailer | 报文末端的首部一览 |
| Transfer-Encoding | 指定报文主体的传输编码方式 |
| Upgrade | 升级为其它协议 |
| Via | 代理服务器的相关信息 |
| Warning | 错误通知 |

表6-2：请求首部字段