# 【01-HTTP协议介绍】

## HTTP 协议介绍

HTTP 协议的主要特点如下。

* 支持Client/Server 模式：
* 简单一一客户向服务器请求服务时，只需指定服务URL，携带必要的请求参数或者消息体：
* 灵活——HTTP 允许传输任意类型的数据对象，传输的内容类型自HTTP 消息头中的Content-Type 加以标记：
* 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之前的信息，则必须重传，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，负载较轻。

## HTTP 协议的URL

HTTP URL ( URL 是一种特殊类型的URI ， 包含了用于查找某个资源的足够的信息）的格式如下。

**http://host:port/path?query#fragment**

其中， http 表示要通过HTTP 协议来定位网络资源： host 表示合法的Internet 主机域名或者IP 地址： port 指定一个端口号，为空则使用默认端口80；path 指定请求资源的URI ，如果URL 中没有给出path ，那么当它作为请求URI 时，必须以“/”的形式给出，通常这点工作浏览器会自动帮我们完成；query是请请求的参数；fragment是请求的片断。

## HTTP 请求消息（HttpRequest）

HTTP 请求由三部分组成，具体如下。

* HTTP 请求行；
* HTTP 消息头：
* HTTP 请求正文。

请求行以一个方法符开头，以空格分开，后面跟着请求的URI 和协议的版本，格式为：

**Method Request-URI HTTP-Version CRLF**

其中Method 表示请求方法， Request-URI 是一个统一资源标识符， HTTP- Version 表示请求的HTTP 协议版本， C RLF 表示回车和换行（除了作为结尾的CRLF 外，不允许出现单独的CR 或LF 字符）。

请求方法有多种，各方法的作用如下。

* + - GET：请求获取Request-URI 所标识的资源：
    - POST ： 在Request-URI 所标识的资源后附加新的提交数据；
    - HEAD：请求按取自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：
    - CONNECT ：保留将来使用：
    - OPTIONS ：请求查询服务榕的性能， 或者查询与资源相关的选项和需求。

### GET & POST

GET 方法：在浏览器的地址栏中输入网址的方式访问网页时， 浏览器采用GET方法向服务器获取资源。POS T 方法要求被请求服务器接受附在请求后面的数据，常用于提交表单。GET 一般用于获取/查询资源信息，而POST 一般用于更新资源信息。GET和POST的主要区别如下。

（1）根据HTTP 规范， G ET 用于信息获取，而且应该是安全的和幂等的； POST 则表示可能改变服务器上的资源的请求。

（2） GET 提交，请求的数据会附在U RL之后，就是把数据放置在请求行（ request line）中，以“?”分隔U RL和传输数据，多个参数用“&”连接： 而POST 提交会把提交的数据放置在HTTP 消息的包体中，数据不会在地址栏中显示出来。

（3）传输数据的大小不同。特定浏览器和服务器对URL 长度有限制，例如IE 对URL长度的限制是2 083 字节（2KB +35B ）， 因此GET 携带的参数的长度会受到浏览器的限制；而POST 由于不是通过URL 传值，理论上数据长度不会受限。

（4）安全性。POST 的安全性要比G ET 的安全性高。比如通过GET 提交数据，用户名和密码将明文出现在URL 上。因为1）登录页面有可能被浏览器缓存；2）其他人查看浏览器的历史记录，那么别人就可以拿到你的账号和密码。除此之外， 使用GET 提交数据还可能会造成Cross- site request forgery 攻击。POST 提交的内容由于在消息体中传输，因此不存在上述安全问题。

请求报头允许客户端向服务器端传递请求的附加信息以及客户端自身的信息。常用的请求表-1，表-2 所示。

HTTP 请求消息体是可选的，比较常用的HTTP+XML 协议就是通过HTTP 请求和响应消息体来承载XML信息的。

## HTTP响应信息