

HTTP协议

简介

- HTTP协议是Hyper Text Transfer Protocol（超文本传输协议）的缩写,是用于从万维网（WWW:World Wide Web）服务器传输超文本到本地浏览器的传送协议。
- HTTP是一个基于TCP/IP通信协议来传递数据（HTML 文件, 图片文件, 查询结果等）。
- 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协议工作于客户端-服务端架构为上。浏览器作为HTTP客户端通过URL向HTTP服务端即WEB服务器发送所有请求。Web服务器根据接收到的请求后，向客户端发送响应信息。



HTTP消息结构

- HTTP是基于客户端/服务端（C/S）的架构模型，通过一个可靠的链接来交换信息，是一个无状态的请求/响应协议。
- 一个HTTP“客户端”是一个应用程序（Web浏览器或其他任何客户端），通过连接到服务器达到向服务器发送一个或多个HTTP的请求的目的。
- 一个HTTP“服务器”同样也是一个应用程序（通常是一个Web服务，如Apache Web服务器或IIS服务器等），通过接收客户端的请求并向客户端发送HTTP响应数据。
- HTTP使用统一资源标识符（Uniform Resource Identifiers, URI）来传输数据和建立连接。一旦建立连接后，数据消息就通过类似Internet邮件所使用的格式[RFC5322]和多用途Internet邮件扩展（MIME）[RFC2045]来传送

客户端请求信息

客户端发送一个HTTP请求到服务器的请求消息包括以下格式：请求行（request line）、请求头部（header）、空行和请求数据四个部分组成

- 一般格式为

请求方法	空格	URL	空格	协议版本	回车符	换行符	请求行
头部字段名	:	值	回车符	换行符	} 请求头部		
...							
头部字段名	:	值	回车符	换行符			
回车符	换行符						
						请求数据	

服务器响应消息

HTTP响应也由四个部分组成，分别是：状态行、消息报头、空行和响应正文。
例如：

HTTP/1.1 200 OK
Date: Sat, 31 Dec 2005 23:59:59 GMT
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 122

<html>
<head>
<title>Wrox Homepage</title>
</head>
<body>
<!-- body goes here -->
</body>
</html>

状态行

消息报头

空行

下面的就是响应正文了

HTTP请求方法

根据HTTP标准，HTTP请求可以使用多种请求方法。

序号	方法	描述
1	GET	请求指定的页面信息，并返回实体主体。
2	HEAD	类似于get请求，只不过返回的响应中没有具体的内容，用于获取报头
3	POST	向指定资源提交数据进行处理请求（例如提交表单或者上传文件）。数据被包含在请求体中。POST请求可能会导致新的资源的建立和/或已有资源的修改。
4	PUT	从客户端向服务器传送的数据取代指定的文档的内容。
5	DELETE	请求服务器删除指定的页面。

6	CONNECT	HTTP/1.1协议中预留给能够将连接改为管道方式的代理服务器。
7	OPTIONS	允许客户端查看服务器的性能。
8	TRACE	回显服务器收到的请求，主要用于测试或诊断。

HTTP状态码

当浏览者访问一个网页时，浏览者的浏览器会向网页所在服务器发出请求。当浏览器接收并显示网页前，此网页所在的服务器会返回一个包含HTTP状态码的信息头（server header）用以响应浏览器的请求。

下面为常见状态码

- 200 - 请求成功
- 301 - 资源（网页等）被永久转移到其它URL
- 404 - 请求的资源（网页等）不存在
- 500 - 内部服务器错误

HTTP状态码的分类

HTTP状态码由三个十进制数字组成，第一个十进制数字定义了状态码的类型，后两个数字没有分类的作用。HTTP状态码共分为5种类型：

- 1** 信息，服务器收到请求，需要请求者继续执行操作
- 2** 成功，操作被成功接收并处理
- 3** 重定向，需要进一步的操作以完成请求
- 4** 客户端错误，请求包含语法错误或无法完成请求
- 5** 服务器错误，服务器在处理请求的过程中发生了错误

HTTP工作原理

- HTTP协议定义Web客户端如何从Web服务器请求Web页面，以及服务器如何把Web页面传送给客户端。HTTP协议采用了请求/响应模型。客户端向服务器发送一个请求报文，请求报文包含请求的方法、URL、协议版本、请求头部和请求数据。服务器以一个状态行作为响应，响应的内容包括协议的版本、成功或者错误代码、服务器信息、响应头部和响应数据。

以下是HTTP请求/响应的步骤

1. 客户端连接到Web服务器
2. 发送HTTP请求
3. 服务器接受请求并返回HTTP响应

4. 释放连接TCP连接
5. 客户端浏览器解析HTML内容

HTTP之URL

- HTTP使用统一资源标识符 (Uniform Resource Identifiers, URI) 来传输数据和建立连接。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
- URL,全称是UniformResourceLocator, 中文叫统一资源定位符,是互联网上用来标识某一处资源的地址。以下面这个URL为例, 介绍下普通URL的各部分组成:

<http://www.aspxfans.com:8080/news/index.asp?boardID=5&ID=24618&page=1#name>

从上面的URL可以看出, 一个完整的URL包括以下几部分:

1. 协议部分: 该URL的协议部分为“http: ”, 这代表网页使用的是HTTP协议。在Internet中可以使用多种协议, 如HTTP, FTP等等本例中使用的是HTTP协议。在“HTTP”后面的“/”为分隔符
2. 域名部分: 该URL的域名部分为“www.aspxfans.com”。一个URL中, 也可以使用IP地址作为域名使用
3. 端口部分: 跟在域名后面的是端口, 域名和端口之间使用“:”作为分隔符。端口不是一个URL必须的部分, 如果省略端口部分, 将采用默认端口
4. 虚拟目录部分: 从域名后的第一个“/”开始到最后一个“/”为止, 是虚拟目录部分。虚拟目录也不是一个URL必须的部分。本例中的虚拟目录是“/news/”
5. 文件名部分: 从域名后的最后一个“/”开始到“?”为止, 是文件名部分, 如果没有“?”,则是从域名后的最后一个“/”开始到“#”为止, 是文件部分, 如果没有“?”和“#”, 那么从域名后的最后一个“/”开始到结束, 都是文件名部分。本例中的文件名是“index.asp”。文件名部分也不是一个URL必须的部分, 如果省略该部分, 则使用默认的文件名
6. 锚部分: 从“#”开始到最后, 都是锚部分。本例中的锚部分是“name”。锚部分也不是一个URL必须的部分
7. 参数部分: 从“?”开始到“#”为止之间的部分为参数部分, 又称搜索部分、查询部分。本例中的参数部分为“boardID=5&ID=24618&page=1”。参数可以允许有多个参数, 参数与参数之间用“&”作为分隔符。

URI和URL的区别

URI, 是uniform resource identifier, 统一资源标识符, 用来唯一的标识一个资源。URL是uniform resource locator, 统一资源定位器, 它是一种具体的URI, 即URL可以用来标识一个资源, 而且还指明了如何locate这个资源。