

# HTTP协议

## HTTP简介

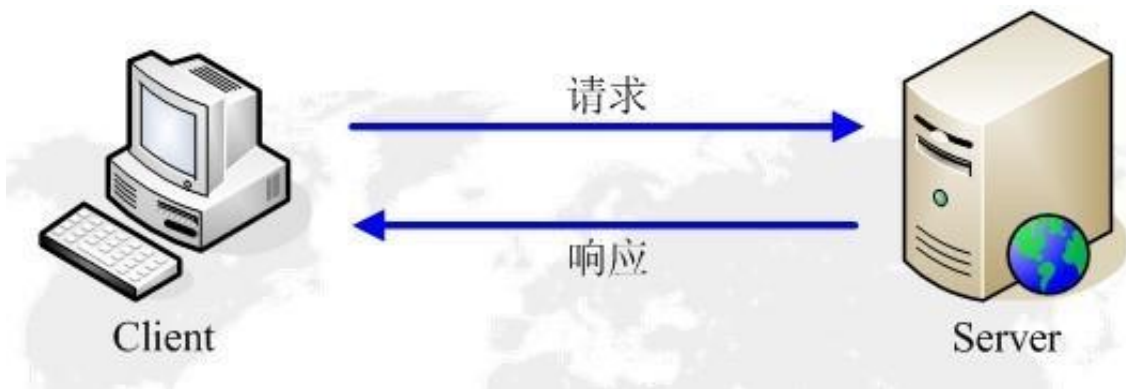
HTTP协议是Hyper Text Transfer Protocol（超文本传输协议）的缩写,是用于从万维网（WWW:World Wide Web）服务器传输超文本到本地浏览器的传送协议。

HTTP协议是一个非常简单的文件上传和下载的协议。支持任意文件类型。

客户端：浏览器，postman 等支持发出http请求的软件。（请求访问文本或图像等资源）

服务端：apache，tomcat等等支持http响应的软件(返回文件)。

HTTP 协议用于客户端和服务端之间的通信。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。



## 主要特点

1、**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。

2、**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。

3.**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**。

4.**无状态**：(既是缺点也是优点)

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。

5、**支持B/S及C/S模式**。

注意：

服务器与客户端可以安装在同一台电脑中，服务器会指定一个本地目录作为网站的根目录。

# HTTP之URL

URL(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“HTTP”后面的“//”为分隔符

http:// 明文传输 不安全 (默认端口号是80)

https:// 密文传输 安全 (默认端口号是433)

2.域名部分: 该URL的域名部为“[www.aspxfans.com](http://www.aspxfans.com)”。一个URL中, 也可以使用IP地址作为域名使用。(一个域名标志着一台主机)

3.端口部分: (表示是哪个软件提供访问, 一台电脑支持多个服务器, 默认有多个端口号) 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: 从端口号后的第一个“/”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。

6.参数部分: 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。

7.锚部分: 从“#”开始到最后, 都是锚部分。本例中的锚部分是“name”。(很少用)

总结下: 一个url就是一个文件, 只是存储在网络中。

## HTTP请求方法

根据HTTP标准, HTTP请求可以使用多种请求方法。HTTP1.0定义了三种请求方法: GET, POST 和 HEAD方法。HTTP1.1新增了五种请求方法: OPTIONS, PUT, DELETE, TRACE 和 CONNECT 方法。

GET	请求指定的页面信息, 并返回实体主体。获取文件, 一般不上传参数。
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。获取文件, 获取文件之前上传参数。
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。 判断文件是否存在
PUT	从客户端向服务器传送的数据取代指定的文档的内容。 创建文件
DELETE	请求服务器删除指定的页面。 删除文件
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。
OPTIONS	允许客户端查看服务器的性能。
TRACE	回显服务器收到的请求, 主要用于测试或诊断。

# HTTP之状态码

状态代码有三位数字组成，第一个数字定义了响应的类别，共分五种类别：

**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

常见状态码：

**200 OK** //客户端请求成功 **400 Bad Request** //客户端请求有语法错误，不能被服务器所理解 **401 Unauthorized** //请求未经授权，这个状态代码必须和WWW-Authenticate报头域一起使用 **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，eg：输入了错误的URL **500 Internal Server Error** //服务器发生不可预期的错误 **503 Server Unavailable** //服务器当前不能处理客户端的请求，一段时间后可能恢复正常

更多状态码<http://www.runoob.com/http/http-status-codes.html>

# HTTP之请求消息Request

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）和请求体(body)三个部分组成。

例如：GET /index.html HTTP/1.1

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

请求行，用来说明请求类型,要访问的资源以及所使用的HTTP版本.请求行以一个方法符号开头，以空格分开，后面跟着请求的URL和协议的版本。

在浏览器中：如何查看 请求和响应的参数内容：

F12 ->network->F5

## ▼ General

请求的响应写在一起

请求路径

Request URL: https://dss0.bdstatic.com/5aV1bjqh\_Q23odCf/static/superman/js/super\_load-2aa20826e0.js

Request Method: GET 请求方法

Status Code: 200 服务器返回的状态

Remote Address: 125.77.154.33:443 远程的连接

Referrer Policy: unsafe-url

- Request Headers就是请求头 (Get方法没有请求体)

## HTTP之响应消息Response

Response Headers就是响应头

一般情况下，服务器接收并处理客户端发过来的请求后会返回一个HTTP的响应消息。

HTTP响应也由三个部分组成，分别是：状态行、消息报头和响应正文。

```
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>
```

Diagram labels:

- 状态行 (Status Line): Points to the first line of the response.
- 消息报头 (Message Header): Points to the header section.
- 空行 (Blank Line): Points to the line separating the header from the body.
- 下面的就是响应正文了 (The following is the response body): Points to the body section.

第一部分：状态行，由HTTP协议版本号，状态码，状态消息 三部分组成。

第一行为状态行，（HTTP/1.1）表明HTTP版本为1.1版本，状态码为200，状态消息为（ok）

第二部分：消息报头，用来说明客户端要使用的一些附加信息

第二行和第三行为消息报头，Date:生成响应的日期和时间；Content-Type:指定了请求文件的类型

这里的文件类型是HTML(text/html),编码类型是ISO-8859-1

content-length:表示要请求内容的长度

第三部分：响应正文，服务器返回给客户端的文本信息。

空行后面的html部分为响应正文。

# form表单提交

输入的url: <http://localhost:8090/login.html>

提交表单 不是提交html, 仅仅是提交表单中input控件 的**name**与**value**的属性值

4、GET与POST的区别:

注意: input控件没有name属性, 不会提交

GET: 请求文件时, 将参数以queryString的方式提交, 而且存在url中



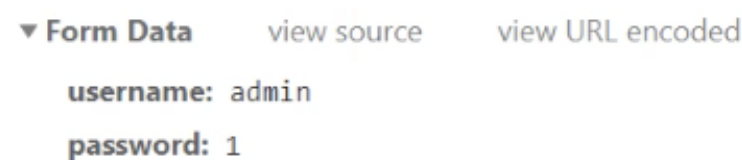
登录失败

组装的方式: action?name=value&name1=value1

queryString是一种**格式**: name=value&name1=value1

POST: 请求文件时, 将参数存放在请求体中的方式提交, 不会存在url中

**Form Data**就是请求体



## 区别重点:

1、参数传递方式不一样

2、**querystring**有长度限制 **form data** 没有长度限制

3、post会比get更安全

因为url有长度限制, 不能超过2KB

只获取数据: get与post随便

还需传递参数: 尽量使用post