

[HTTP简介](#)  
[主要特点](#)  
[HTTP之URL](#)  
[URL 于 URI 的区别](#)  
[HTTP之Request](#)  
[HTTP之响应消息Response](#)  
[HTTP之状态码](#)  
[HTTP工作原理](#)  
[GET 和 POST请求的区别](#)

## HTTP简介

HTTP: **Hyper Text Transfer Protocol** (超文本传输协议)

基于TCP/IP通信协议来传递数据

### 主要特点

- 1、简单快速：请求时只需传送请求方法（GET\POST\HEAD)和路径
- 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标记
- 3、无连接：不建立一个固定的连接，只对应每次的请求建立连接，请求完毕时连接断开，这样节省时间
- 4、无状态：和无连接逻辑相同，不需要记忆之前的请求状态。即如果后续处理需要前面的信息必须重传
- 5、支持B/S 和 C/S

B/S 和 C/S 结构区别：

- C/S: Client / Server 结构。将任务分配给C ,S 两端，充分利用两端硬件环境的优势
- B/S: Brower / Server结构。对C/S结构的一种变化或者改进的结构。此结构下用户界面通过Browser实现，主要事务逻辑在server实现，部分逻辑借用多种脚本语言在本地前端实现。
- 具体区别：
- 硬件环境不同
  - C/S 建立在局域网基础上,B/S建立在广域网基础上。
  - C/S一般使用专用网络, B/S对网络硬件环境无特殊要求。只需要操作系统+浏览器
- 对安全要求不同
  - C/S 对两端都考虑安全性
  - B/S 只需要注重服务器端
- 程序架构不同
  - C/S 相比速度更注重安全（权限）问题
  - B/S 由于安全性只需要考虑服务器端自身，更注重的是优化程序，提高访问速度
- 软件重用不同
  - C/S 软件重用性 低于 B/S
- 系统维护不同
  - C/S 需要顾及两端

## HTTP之URL

## URI:Uniform Resource Identifiers：统一资源标识符

URL: **Uniform Resource Locator** : 统一资源定位符

例如: <https://cn.bing.com/search?q=http%e5%8d%8f%e8%ae%ae&q=AS&pg=http%e5%8d%8f%e8%ae%ae&sc=9-6&cvid=6817DD8B700A466CAA50A326A9A5DCA3&FORM=QBLH&sp=2>

or:

<https://www.cnblogs.com/ranyonsue/p/5984001.html>

## URL组成

1. 协议部分：URL开头表明使用HTTP/HTTPS/FTP, 协议后：// 表示分隔符
2. 域名+端口：端口和域名间用‘:’ 隔开，不显示端口即使用默认端口
3. 虚拟目录：如上述二例中的‘ranyonsue’
4. 文件名：从域名后的最后一个‘/’ 开始到‘?’（或者‘#’）为止为所访问的文件名。当没有‘?’ 和‘#’ 时, 从最后一个‘/’ 到结尾都是文件名。即最后一级访问文件。同时若无，则是使用默认文件例如：[www.bing.com](http://www.bing.com)
5. 锚部分：从‘#’ 开始到最后，为锚
6. 参数部分：从‘?’ 到‘#’ 之间的部分都为参数，参数与参数之间采用‘&’ 分隔

## URL 于 URI 的区别

## HTTP之Request

一个HTTP请求包括：

请求行+请求头部+空行+请求数据

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	

例如：

▼ 请求标头

```
:authority: data.bilibili.com
:method: POST
:path: /log/web?0000161590133106119https%3A%2F%2Fwww.bilibili.com%2Fvideo%2FBV1i7411e7CY%3Ffrom%3Dsearch%26seid%3D3130241334940944712|333.788.comment.0|1590137430975|1463x788|1|{%22num%22:6,%22name%22:%22report-scroll-module%22}|{%22hitGroup%22:%22863_658%22}
:scheme: https
accept: */*
accept-encoding: gzip, deflate, br
accept-language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
content-length: 0
content-type: text/plain;charset=UTF-8
cookie: _uuid=50CEA33E-0492-D853-45A5-E5C916D5272488438infoc; buvid3=243502B3-FB86-4A76-96F9-8D9CCF470B52155835infoc; sid=6vdf7580; CURRENT_FNVAL=16; LIVE_BUVID=AUTO3415829617276925; rpidid=|(k|~JJm|Juu0J'u1)kmYu~JY; im_notify_type_249390129=0; CURRENT_QUALITY=80; bp_t_offset_372270361=381388856756555409; DedeUserID=249390129; DedeUserID_ckMd5=3788d6b10f5200d9; SESSDATA=d5bc2f3b%2C1603530047%2C2f6a1*41; bili_jct=52db624b8611fb677d8fa601d6109173; bp_t_offset_249390129=385520177207067237; bp_video_offset_249390129=391723802203320663; PVID=1; _dfcaptcha=d213979a63ec98a0615962527451dc5f
origin: https://www.bilibili.com
referer: https://www.bilibili.com/video/BV1i7411e7CY?p=2
sec-fetch-dest: empty
sec-fetch-mode: no-cors
sec-fetch-site: same-site
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.
```

POST / HTTP1.1 ← ①

Host:www.wrox.com

User-Agent:Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022)

Content-Type:application/x-www-form-urlencoded

Content-Length:40

Connection: Keep-Alive

name=Professional%20Ajax&publisher=Wiley ← ④

①：请求行

②：请求头部

③：空行

④：请求数据

## HTTP之响应消息Response

Response四个部分组成：状态行、消息报头、空行、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>
```

状态行

消息报头

空行

下面的就是响应正文了

状态行：HTTP协议版本号+状态码+状态消息 三部分

消息报头：说明客户端需要使用的一些附加消息

- Date：响应的日期和时间
- Content - Type
- 编码类型
- 长度

空行：必须的

响应正文：服务器返还给客户端的文本信息；html语言

## HTTP之状态码

状态码均为三个数字，第一个数字定义响应类别，共五种

1xx:指示信息，表示请求已接收正在处理

2xx:接收成功，表示请求被成功接收、理解、接受

3xx: 重定向，若完成请求需要更进一步操作

4xx: 客户端错误，请求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请求

?

下面是常见的HTTP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

状态码集合：<https://www.runoob.com/http/http-status-codes.html>

# HTTP工作原理

1. 客户端连接到web服务器；一个HTTP客户端通常是浏览器和服务器的HTTP端口建立一个TCP套接字连接
2. 发送HTTP请求：通过TCP套接字，客户端向服务器发送一个文本的请求报文（四部分：请求行、请求头部、空行、数据）
3. 服务器接受请求并返回HTTP响应：服务器解析请求，并复写到TCP套接字中让客户端读取。响应报文组成：响应行、头部、空行、响应数据
4. 释放连接TCP连接:由HTTP协议的Connection 状态决定如何释放TCP连接。若connection模式为close 则服务器主动关闭TCP连接，客户端被动关闭；若connection模式为keepalive则保持该连接一段时间，此段时间内可持续接受请求。
5. 客户端浏览器解析HTML内容:先查看响应状态行，再解析每个响应头，再读取响应数据的HTML语言，格式化在本地显示

- 1、浏览器向 DNS 服务器请求解析该 URL 中的域名所对应的 IP 地址;
- 2、解析出 IP 地址后，根据该 IP 地址和默认端口 80，和服务器建立[TCP连接](#);
- 3、浏览器发出读取文件(URL 中域名后面部分对应的文件)的HTTP 请求，该请求报文作为 [TCP 三次握手](#)的第三个报文的数据发送给服务器;
- 4、服务器对浏览器请求作出响应，并把对应的 html 文本发送给浏览器;
- 5、释放 [TCP连接](#);
- 6、浏览器将该 html 文本并显示内容;

## GET 和 POST请求的区别

GET方式请求数据附在URL之后，由于URL的长度受限，所以GET请求数据长度也受限。POST则是放在HTTP包的包体中，不受URL长度的限制。同时POST方式将请求数据不直接在URL栏中显示，相比GET方式更加安全（伪安全）

参考文献：博客园:<https://www.cnblogs.com/ranyonsue/p/5984001.html>