

HTTP协议

▼ Status	
≡ Author	
📅 Publishing/Release Date	
▼ Publisher	
🔗 Link	
≡ Summary	
▼ Score /5	
▼ Type	intern

概述

- 超文本传输协议
- 基于请求和响应
- 无状态
- 应用层协议
- 基于TCP/IP传输数据

发展历史

- HTTP/0.9, 1991, 只能GET请求, 非正式标准
- HTTP/1.0 , 1996, PUT\PATCH\HEAD\OPTIONS\DELETE, 正式标准；目前使用最多的版本
- HTTP/1.1, 1997, 支持长连接
- HTTP/2, 2015, 多路复用, 头信息压缩

特点

- 无状态
协议对客户端没有状态存储, 对事物处理没有“记忆”能力
- 无连接
1.1版本前, 每次请求需要通过TCP三次握手四次挥手, 和服务器重新建立连接

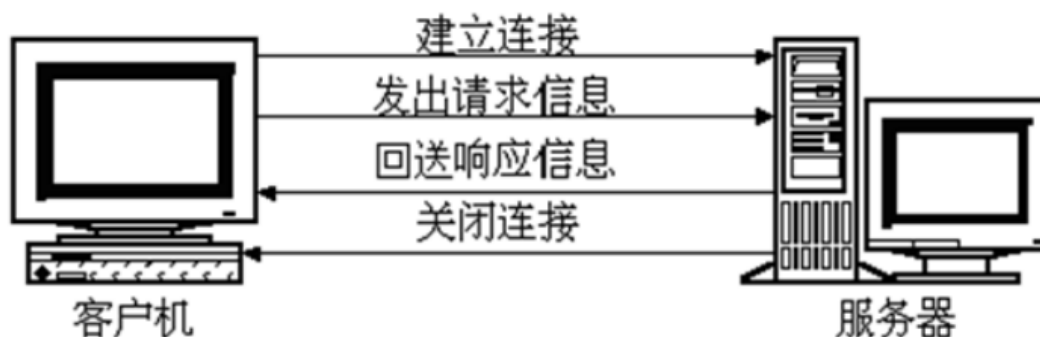
- 基于请求和响应
客户端发起请求，服务端响应
- 明文通信
请求和响应不会对通信方进行确认、无法保护数据的完整性
- 状态保持
Cookie Session JWT
- 80端口

HTTPS

- 身披SSL(TLS)外壳的HTTP
- 网站服务器的身份认证，保护交换数据的隐私与完整性
- 内容加密
中间者无法直接查看明文内容
- 身份验证
通过证书认证客户端访问的是自己的服务器
- 数据完整性
防止传输的内容被中间人冒充或者篡改
- 443端口

HTTP协议的会话方式

- 浏览器与服务器之间的通信过程要经历**四个步骤**

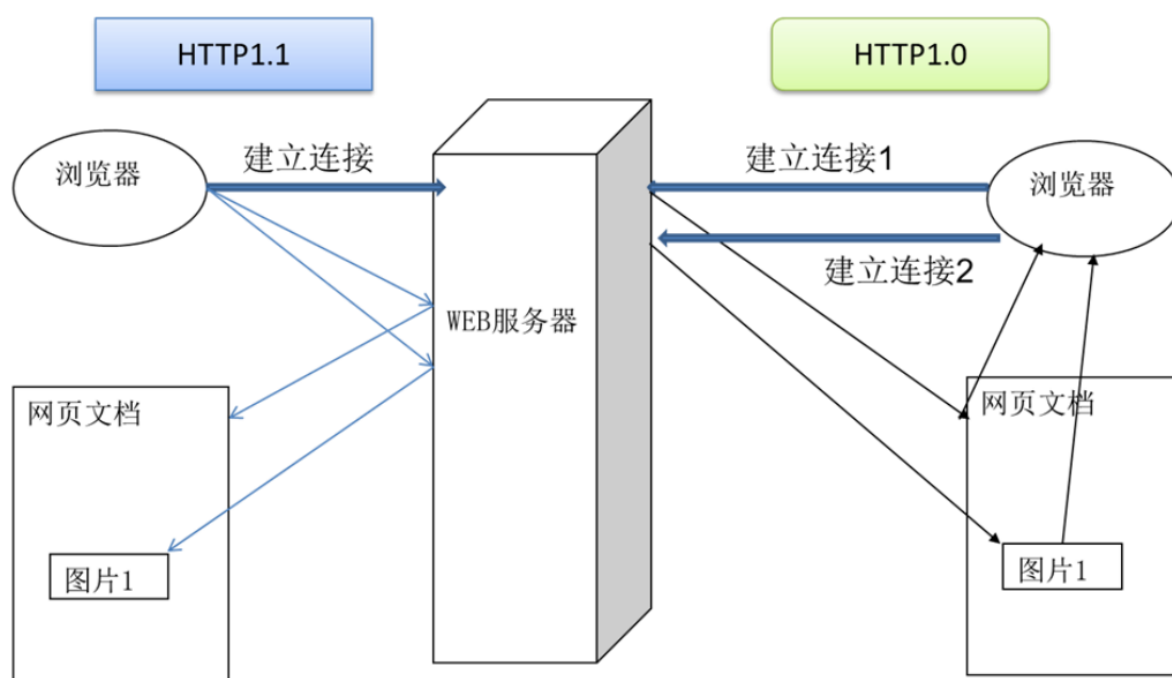


- 浏览器与 WEB 服务器的连接过程是短暂的，每次连接只处理一个请求和响应。对每一个页面的访问，浏览器与 WEB 服务器都要建立一次单独的连接。

- 浏览器到 WEB 服务器之间的所有通讯都是完全独立分开的请求和响应对。

HTTP1.0和HTTP1.1的区别

在 HTTP1.0 版本中，浏览器请求一个带有图片的网页，会由于下载图片而与服务器之间开启一个新的连接；但在 HTTP1.1 版本中，允许浏览器在拿到当前请求对应的全部资源后再断开连接，提高了效率。



报文

- 报文格式

报文首部	【 报文首部 】 服务器端或客户端需处理的请求或响应的内容及属性
空行(CR+LF)	【 CR + LF 】 CR(Carriage Return, 回车符: 16进制 0x0d)和 LF(Line Feed, 换行符: 16进制 0x0a)
报文主体	【 报文主体 】 应被发送的数据

- 请求报文

- 报文格式

```
请求首行（请求行）；
请求头信息（请求头）；
空行；
请求体；
```

- GET请求

```
GET /Hello/index.jsp HTTP/1.1
Accept: */*
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64;
Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
Media Center PC 6.0; .NET4.0C; .NET4.0E)
Accept-Encoding: gzip, deflate
Host: localhost:8080
Connection: Keep-Alive
Cookie: JSESSIONID=C55836CDA892D9124C03CF8FE8311B15
```

- **Get 请求一般没有请求体**
- GET /Hello/index.jsp HTTP/1.1 ： GET 请 求 ， 请 求 服 务 器 路 径 为 Hello/index.jsp，协议为 1.1 ；
- Host:localhost：请求的主机名为 localhost ；
- User-Agent: Mozilla/4.0 (compatible; MSIE 8.0...：与浏览器和 OS 相关的信息。有些网站会显示用户的系统版本和浏览器版本信息，这都是通过获取 UserAgent 头信息而来的 ；

- Accept: /：告诉服务器，当前客户端可以接收的文档类型， /，就表示什么都可以接收
- Accept-Language: zh-CN：当前客户端支持的语言，可以在浏览器的工具选项中找到语言相关信息
- Accept-Encoding: gzip, deflate：支持的压缩格式。数据在网络上传递时，可能服务器会把数据压缩后再发送
- Connection: keep-alive：客户端支持的链接方式，保持一段时间链接，默认为 3000ms
- Cookie: JSESSIONID=369766FDF6220F7803433C0B2DE36D98：因为不是第一次访问这个地址，所以会在请求中把上一次服务器响应中发送过来的 Cookie在请求中一并发送过去

○ POST请求

```
POST /Hello/target.html HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml,
image/gif, image/pjpeg, application/x-ms-xbap, */*
Referer: http://localhost:8080/Hello/
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64;
Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729;
Media Center PC 6.0; .NET4.0C; .NET4.0E)
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: localhost:8080
Content-Length: 14
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: JSESSIONID=774DA38C1B78AE288610D77621590345
username=admin
```

- Referer: <http://localhost:8080/hello/index.jsp>：请求来自哪个页面，例如你在百度上点击链接到了这里，那么 Referer:<http://www.baidu.com>；如果你是在浏览器的地址栏中直接输入的地址，那么就没有 Referer 这个请求头了
- Content-Type: application/x-www-form-urlencoded：表单的数据类型，说明会使用 url 格式编码数据；url 编码的数据都是以“%”为前缀，后面跟

随两位的 16 进制，例如“传智”这两个字使用 UTF-8 的 url 编码用为“%E4%BC%A0%E6%99%BA”；

- Content-Length:13：请求体的长度，这里表示 13 个字节
- keyword=hello：请求体内容！hello 是在表单中输入的数据，keyword 是表单字段的名字

- 响应报文

- 报文格式

响应首行（响应行）；
响应头信息（响应头）；
空行；
响应体；

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=UTF-8
Content-Length: 274
Date: Tue, 07 Apr 2015 10:08:26 GMT
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
<h1>Hello</h1>
</body>
</html>
```

- HTTP/1.1 200 OK：响应协议为 HTTP1.1，状态码为 200，表示请求成功
- Server: Apache-Coyote/1.1：服务器的版本信息
- Content-Type: text/html;charset=UTF-8：响应体使用的编码为 UTF-8
- Content-Length: 274：响应体为 274 字节；
- Date: Tue, 07 Apr 2015 10:08:26 GMT：响应的时间，这可能会有 8 小时时区差

响应状态码

- 状态码用来告诉 HTTP 客户端,HTTP 服务器是否产生了预期的 Response。

HTTP/1.1

协议中定义了 5 类状态码， 状态码由三位数字组成， 第一个数字定义了响应的类别

- 1XX 提示信息 - 表示请求已被成功接收，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处理
 - 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
- 响应码对浏览器来说很重要，它告诉浏览器响应的结果，常见的状态码有
 - 200：请求成功，浏览器会把响应体内容（通常是 html）显示在浏览器中
 - 404：请求的资源没有找到，说明客户端错误的请求了不存在的资源
 - 500：请求资源找到了，但服务器内部出现了错误
 - 302：重定向，当响应码为 302 时，表示服务器要求浏览器重新再发一个请求，
服务器会发送一个响应头 Location，它指定了新请求的 URL 地址