



链滴

Http 协议

作者: [whoms](#)

原文链接: <https://ld246.com/article/1640254219523>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

1. Http协议

• 概念

在BS架构中，浏览器通过Http协议访问服务器。它是一种应用层协议，运行在TCP传输层协议之上
Http作用是指定数据的传输格式。

默认端口号：80

• Http协议特点

- Http协议组成：由请求和响应组成
- 是一种无状态协议，不记录用户访问状态。同一个用户在同一浏览器上发送多次请求，服务器并不能通过Http协议判断是否是同一个用户
- 传输效率相对较高

2. Http请求

• Http请求的组成

• 请求行

• 请求的方式

- Get
- Post

• 请求的地址

• 协议和版本

- 1.0版本：每次请求都会关闭TCP连接
- 1.1版本：普遍使用版本，一次请求多个web资源，不会在每次请求结束关闭TCP连接

• 请求头（不是每次请求都会显示所有的请求头，常用）

• **Referer**：表示从哪个页面跳转到当前页面来的，得到上一个访问的页面。可以通过该属性设防盗链，通过判断上一个页面来决定是否允许访问该页面

• **User-Agent**：表示当前客户端浏览器的类型信息

• **Connection**：当前服务器连接状态

• keep-alive：保持连接状态

• close：连接关闭

• **Host**：连接的服务器名字和端口

• ...

• 请求体（Get方法发送数据时没有请求体）

- 本次请求携带的用户数据

• 请求方式的对比 | POST方式 | GET方式

地址栏	参数不会在地址栏上显示出来	参数会显示出来，以查询字符串的方式发送数据
大小	理论上没有大小限制，使用流的方式来发送数据。	受浏览器限制，通常是1K
安全性	POST数据在请求体中发送，安全性要高一些	数据在请求行中发送，安全性要低一些
缓存	不使用缓存 存，如果使用缓存状态码是304	如果静态资源在浏览器端已经访问过，下次可能会使用

• HttpServletRequest

• 是ServletRequest接口的子接口，代表一个请求对象，用来封装所有从浏览器发送给服务器的数据

• 获取请求行相关方法

- `String getMethod()`: 得到请求的方式
- `String getRequestURI`: 得到请求的URI，只是地址的一部分
- `StringBuffer getRequestURL()`: 得到请求的URL地址
- `String getProtocol()`: 得到请求行中的协议和版本
- `String getRemoteAddr()`: 得到客户端的IP地址
- `String getContextPath()`: 得到当前项目的访问地址
- `String getServletPath()`: 得到当前Servlet的访问地址

• 获取请求头相关方法

请求头中是由各种键值对组成，通过键得到值

- `String getHeader(String headName)`: 通过键的名字得到对应的值
- `Enumeration<String> getHeaderNames()`: 得到所有请求头中键的名字，返回枚举类型

• 获取请求参数

- `String getParameter(String name)`: 通过参数名字得到参数值
- `String[] getParameterValues(String name)`: 得到所有同名的参数的值，一般用于复选框
- `Enumeration<String> getParameterNames()`: 得到所有的参数名字，返回枚举类型
- `Map<String,String[]> getParameterMap()`: 得到所有的参数名和值，封装为Map对象

• BeanUtils的使用

BeanUtils是Apache Commons组件的成员之一，主要用于简化JavaBean封装数据的操作

- 下载地址: <http://commons.apache.org/>
- 相关jar包
 - `commons-beanutils-1.9.3.jar`: BeanUtils工具的核心包

- commons-logging-1.2.jar: 用于日志记录功能
 - commons-collections-3.2.2.jar: 公共集合增强包
- 常用方法
 - public static void populate(Object bean, Map<String,? extends Object> properties)
将Map中所有的键和值封装到bean的同名的属性中，如果不同的名字将忽略。
- 参数的乱码问题
 - 请求参数产生乱码的原因

Tomcat中默认对使用Post方法提交的参数使用ISO-8859-1编码，这是一个欧洲码表，不支持中文

- Post方式乱码解决方案
 - request.setCharacterEncoding("utf-8"), 告诉Tomcat, post方法使用utf-8进行编码
 - 代码必须放在获取参数之前
 - 必须与页面的编码相同
- 关于Tomcat8.0中Get方法乱码问题

Tomcat8.0以后，Get方法参数是在地址栏发送，那么默认就是utf-8编码，所以Get方法汉字没有乱码问题。只能做字符传递。

3. Http响应

- 响应组成
 - 响应行
 - 协议和版本
 - 状态码（根据状态码可以定位错误）
 - 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
 - 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: 服务端错误，服务器未能实现合法的请求
 - 状态信息
- 响应头
 - Location: 重定向要跳转的地址
 - Server: 访问的服务器名字
 - Content-Encoding: 服务器的压缩格式，如果服务器端数据量大时，首先会将数据进行压缩，这就是指压缩的格式，浏览器根据这个格式进行解压
 - Content-Length: 响应的长度
 - Content-Type: 服务器发送过来的数据类型和字符集

- Refresh: 表示n秒以后, 跳转到某个页面
 - Content-Disposition: 内容处理的方式
 - attachment: 以附件的方式进行下载
 - filename: 下载时的文件名, 汉字将乱码
 - Date: 服务器响应时间
 - expires: 浏览器回送的资源存多长时间 (0或-1表示不缓存)
- 响应体
 - 响应的信息
- 常见状态码的具体含义

状态码	含义
200	正确的从服务器得到响应的数据
302	表示页面重定向
304	使用的是本地缓存, 并没有从服务器上再次
载网页	
404	找不到服务器上指定的资源
405	如果客户端使用GET/POST方法提交, 而服
器端没有doGet/doPost方法, 就会出现405错误	
415	服务器无法处理请求附带的媒体格式 (无法
别请求中携带的参数)	
500	服务器出现内部错误, 一般是Servlet中的
码有错误	

- HttpServletResponse
 - 设置状态码: `setStatus(int status)`
 - 设置响应头
 - `void setHeader(String name,String value)`: 设置响应头, 指定名字和值
 - `void setContentType(String type)`: 设置响应的内容
 - `void setCharacterEncoding(String charset)`: 设置响应的字符集, 设置打印流编码
 - 响应体方法
 - `OutputStream getOutputStream()`: 字节输出流, 用于向浏览器输出图片、视频、压缩包
 - `PrintWriter getWrite()`: 字符打印输出流, 默认指向请求的客户端, 用于输出文本数据
- 常见的ContentType类型

请求和响应都有该类型

Content-Type对照表: <https://tool.oschina.net/commons>

text/html	HTML格式
Content-Type类型	说明
text/plain	纯文本格式（默认的）
text/xml	XML格式
image/gif	gif图片格式
image/jpeg	jpg图片格式
image/png	png图片格式
application/xhtml+xml	XHTML格式
application/xml	XML数据格式
application/json	JSON数据格式
application/pdf	pdf格式
application/msword	Word文档格式
application/octet-stream	二进制流数据（如
见的文件下载）	
application/x-www-form-urlencoded	表单
默认的encType, form表单数据被编码为key/value格式发送到服务器（表单默认的提交数据的格式）	