

# Servlet 客户端 HTTP 请求

当浏览器请求网页时，它会向 Web 服务器发送特定信息，这些信息不能被直接读取，因为这些信息是作为 HTTP 请求的头的一部分进行传输的。您可以查看 [HTTP 协议](#) 了解更多相关信息。

以下是来自于浏览器端的重要头信息，您可以在 Web 编程中频繁使用：

头信息	描述
Accept	这个头信息指定浏览器或其他客户端可以处理的 MIME 类型。值 <b>image/png</b> 或 <b>image/jpeg</b> 是最常见的两种可能值。
Accept-Charset	这个头信息指定浏览器可以用来显示信息的字符集。例如 ISO-8859-1。
Accept-Encoding	这个头信息指定浏览器知道如何处理的编码类型。值 <b>gzip</b> 或 <b>compress</b> 是最常见的两种可能值。
Accept-Language	这个头信息指定客户端的首选语言，在这种情况下，Servlet 会产生多种语言的结果。例如，en、en-us、ru 等。
Authorization	这个头信息用于客户端在访问受密码保护的网页时识别自己的身份。
Connection	这个头信息指示客户端是否可以处理持久 HTTP 连接。持久连接允许客户端或其他浏览器通过单个请求来检索多个文件。值 <b>Keep-Alive</b> 意味着使用了持续连接。
Content-Length	这个头信息只适用于 POST 请求，并给出 POST 数据的大小（以字节为单位）。
Cookie	这个头信息把之前发送到浏览器的 cookies 返回到服务器。
Host	这个头信息指定原始的 URL 中的主机和端口。
If-Modified-Since	这个头信息表示只有当页面在指定的日期后已更改时，客户端想要的页面。如果没有新的结果可以使用，服务器会发送一个 304 代码，表示 <b>Not Modified</b> 头信息。
If-Unmodified-Since	这个头信息是 If-Modified-Since 的对立面，它指定只有当文档早于指定日期时，操作才会成功。
Referer	这个头信息指示所指向的 Web 页的 URL。例如，如果您在网页 1，点击一个链接到网页 2，当浏览器请求网页 2 时，网页 1 的 URL 就会包含在 Referer 头信息中。
User-Agent	这个头信息识别发出请求的浏览器或其他客户端，并可以向不同类型的浏览器返回不同的内容。

# 读取 HTTP 头的方法

下面的方法可用在 Servlet 程序中读取 HTTP 头。这些方法通过 *HttpServletRequest* 对象可用。

序号 方法 & 描述	
1	<b>Cookie[] getCookies()</b> 返回一个数组，包含客户端发送该请求的所有的 Cookie 对象。
2	<b>Enumeration getAttributeNames()</b> 返回一个枚举，包含提供给该请求可用的属性名称。
3	<b>Enumeration getHeaderNames()</b> 返回一个枚举，包含在该请求中包含的所有的头名。
4	<b>Enumeration getParameterNames()</b> 返回一个 String 对象的枚举，包含在该请求中包含的参数的名称。
5	<b>HttpSession getSession()</b> 返回与该请求关联的当前 session 会话，或者如果请求没有 session 会话，则创建一个。
6	<b>HttpSession getSession(boolean create)</b> 返回与该请求关联的当前 HttpSession，或者如果没有当前会话，且创建是真的，则返回一个新的 session 会话。
7	<b>Locale getLocale()</b> 基于 Accept-Language 头，返回客户端接受内容的首选的区域设置。
8	<b>Object getAttribute(String name)</b> 以对象形式返回已命名属性的值，如果没有给定名称的属性存在，则返回 null。
9	<b>ServletInputStream getInputStream()</b> 使用 ServletInputStream，以二进制数据形式检索请求的主体。
10	<b>String getAuthType()</b> 返回用于保护 Servlet 的身份验证方案的名称，例如，"BASIC" 或 "SSL"，如果JSP没有受到保护则返回 null。
11	<b>String getCharacterEncoding()</b> 返回请求主体中使用的字符编码的名称。
12	<b>String getContentType()</b> 返回请求主体的 MIME 类型，如果不知道类型则返回 null。
13	<b>String getContextPath()</b> 返回指示请求上下文的请求 URI 部分。

14	<b>String getHeader(String name)</b> 以字符串形式返回指定的请求头的值。
15	<b>String getMethod()</b> 返回请求的 HTTP 方法的名称，例如，GET、POST 或 PUT。
16	<b>String getParameter(String name)</b> 以字符串形式返回请求参数的值，或者如果参数不存在则返回 null。
17	<b>String getPathInfo()</b> 当请求发出时，返回与客户端发送的 URL 相关的任何额外的路径信息。
18	<b>String getProtocol()</b> 返回请求协议的名称和版本。
19	<b>String getQueryString()</b> 返回包含在路径后的请求 URL 中的查询字符串。
20	<b>String getRemoteAddr()</b> 返回发送请求的客户端的互联网协议（IP）地址。
21	<b>String getRemoteHost()</b> 返回发送请求的客户端的完全限定名称。
22	<b>String getRemoteUser()</b> 如果用户已通过身份验证，则返回发出请求的登录用户，或者如果用户未通过身份验证，则返回 null。
23	<b>String getRequestURI()</b> 从协议名称直到 HTTP 请求的第一行的查询字符串中，返回该请求的 URL 的一部分。
24	<b>String getRequestedSessionId()</b> 返回由客户端指定的 session 会话 ID。
25	<b>String getServletPath()</b> 返回调用 JSP 的请求的 URL 的一部分。
26	<b>String[] getParameterValues(String name)</b> 返回一个字符串对象的数组，包含所有给定的请求参数的值，如果参数不存在则返回 null。
27	<b>boolean isSecure()</b> 返回一个布尔值，指示请求是否使用安全通道，如 HTTPS。
28	<b>int getContentLength()</b>

	以字节为单位返回请求主体的长度，并提供输入流，或者如果长度未知则返回 -1。
29	<b>int getIntHeader(String name)</b> 返回指定的请求头的值为一个 int 值。
30	<b>int getServerPort()</b> 返回接收到这个请求的端口号。
31	<b>int getParameterMap()</b> 将参数封装成 Map 类型。

## HTTP Header 请求实例

下面的实例使用 `HttpServletRequest` 的 `getHeaderNames()` 方法读取 HTTP 头信息。该方法返回一个枚举，包含与当前的 HTTP 请求相关的头信息。

一旦我们有一个枚举，我们可以以标准方式循环枚举，使用 `hasMoreElements()` 方法来确定何时停止，使用 `nextElement()` 方法来获取每个参数的名称。

```
//导入必需的 java 库
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/DisplayHeader")

//扩展 HttpServlet 类
public class DisplayHeader extends HttpServlet {

    // 处理 GET 方法请求的方法
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
    IOException
    {
        // 设置响应内容类型
        response.setContentType("text/html;charset=UTF-8");

        PrintWriter out = response.getWriter();
        String title = "HTTP Header 请求实例 - 菜鸟教程实例";
        String docType =
            "<!DOCTYPE html> \n";
```

```
        out.println(docType +
        "<html>\n" +
        "<head><meta charset=\"utf-8\"><title>" + title + "</title></head>\n"+
        "<body bgcolor=\"#f0f0f0\">\n" +
        "<h1 align=\"center\">" + title + "</h1>\n" +
        "<table width=\"100%\" border=\"1\" align=\"center\">\n" +
        "<tr bgcolor=\"#949494\">\n" +
        "<th>Header 名称</th><th>Header 值</th>\n"+
        "</tr>\n");

        Enumeration headerNames = request.getHeaderNames();

        while(headerNames.hasMoreElements()) {
            String paramName = (String)headerNames.nextElement();
            out.print("<tr><td>" + paramName + "</td>\n");
            String paramValue = request.getHeader(paramName);
            out.println("<td> " + paramValue + "</td></tr>\n");
        }
        out.println("</table>\n</body></html>");
    }
    // 处理 POST 方法请求的方法
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException
    , IOException {
        doGet(request, response);
    }
}
```

以上测试实例是位于 TomcatTest 项目下，对应的 web.xml 配置为：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <servlet>
        <!-- 类名 -->
        <servlet-name>DisplayHeader</servlet-name>
        <!-- 所在的包 -->
        <servlet-class>com.runoob.test.DisplayHeader</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>DisplayHeader</servlet-name>
        <!-- 访问的网址 -->
        <url-pattern>/TomcatTest/DisplayHeader</url-pattern>
    </servlet-mapping>
</web-app>
```

现在，调用上面的 Servlet，访问 <http://localhost:8080/TomcatTest/DisplayHeader> 会产生以下结果：

# HTTP Header 请求实例 - 菜鸟教程实例

Header 名称	Header 值
host	localhost:8080
connection	keep-alive
cache-control	max-age=0
upgrade-insecure-requests	1
user-agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
accept-encoding	gzip, deflate, sdch
accept-language	zh-CN,zh;q=0.8,en;q=0.6,zh-TW;q=0.4
cookie	JSESSIONID=60A8DBEC591EC628B0C572AA5EBF5DD9; CNZZDATA1254569789=1732093401-1466493487-%7C1468220728; pgv_pvid=1015625625

[← Servlet 表单数据](#)[Servlet 服务器 HTTP 响应 →](#)[✎ 点我分享笔记](#)