



链滴

SystemUtil Java 工具类 获取 HttpServletRequest request 中的访问者的一些信息

作者: [y kz200](#)

原文链接: <https://ld246.com/article/1498020646713>

来源网站: [链滴](#)

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

本文由[黑壳网](#)原创

文章来源[SystemUtil工具类 获取HttpServletRequest中的访问者的一些信息 - 黑壳网](#)

###壳叔搞笑一刻

####追我玩

一天深夜，一个年轻女子经过一家精神病院时，突然后面传来哇的一声。

女子扭头一看，一个一丝不挂的男子正在向她追来。

女子吓得拔腿就跑，后面的男人紧追不舍。

不好，前面是一条死胡同，女子万念俱灰，跪在地上哭着哀求道:你愿意干什么就干什么吧，只求你不杀我。

男子狡黠地笑了笑说：真的？那现在你开始追我。

通过HttpServletRequest获取ip地址，浏览器版本，以及所使用的系统型号，以及所访问的网址。

其实代码非常简单，都是一些基础的，只是我给整理了一下，可以拿过去直接用。

systemUtil.java

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
```

```
import javax.servlet.http.HttpServletRequest;
import java.net.InetAddress;
import java.net.UnknownHostException;
```

```
/**
```

```
 * Created by kzyuan on 2017/6/21. */public class SystemUtils {
```

```
    private static Logger logger = LoggerFactory.getLogger(SystemUtils.class);
```

```
/**
```

```
 * 获取访问者IP * 在一般情况下使用Request.getRemoteAddr()即可，但是经过nginx等反向代理软件后，这个方法会失效。 *
```

```
 * 本方法先从Header中获取X-Real-IP，如果不存在再从X-Forwarded-For获得第一个IP(用,分割)，
```

```
 * 如果还不存在则调用Request.getRemoteAddr()。 * * @param request
```

```
 * @return
```

```
 */
```

```
public static String getIpAddr(HttpServletRequest request) {
```

```
    String ip = request.getHeader("X-Real-IP");
```

```
if (ip != null && !" ".equals(ip) && !"unknown".equalsIgnoreCase(ip)) {
    return ip;
}
```

```
    ip = request.getHeader("X-Forwarded-For");
```

```
if (ip != null && !" ".equals(ip) && !"unknown".equalsIgnoreCase(ip)) {
    // 多次反向代理后会有多个IP值，第一个为真实IP。
```

```
    int index = ip.indexOf(',');
```

```
if (index != -1) {
```

```
    return ip.substring(0, index);
```

```
    } else {
```

```
        return ip;
```

```
    }
```

```

    } else {
        return request.getRemoteAddr();
    }
}

/**
 * 获取来访者的浏览器版本 * * @param request
 * @return
 */
public static String getRequestBrowserInfo(HttpServletRequest request) {
    String browserVersion = null;
    String header = request.getHeader("user-agent");
    if (header == null || header.equals("")) {
        return "";
    }
    if (header.indexOf("MSIE") > 0) {
        browserVersion = "IE";
    } else if (header.indexOf("Firefox") > 0) {
        browserVersion = "Firefox";
    } else if (header.indexOf("Chrome") > 0) {
        browserVersion = "Chrome";
    } else if (header.indexOf("Safari") > 0) {
        browserVersion = "Safari";
    } else if (header.indexOf("Camino") > 0) {
        browserVersion = "Camino";
    } else if (header.indexOf("Konqueror") > 0) {
        browserVersion = "Konqueror";
    }
    return browserVersion;
}

/**
 * 获取系统版本信息 * * @param request
 * @return
 */
public static String getRequestSystemInfo(HttpServletRequest request) {
    String systemInfo = null;
    String header = request.getHeader("user-agent");
    if (header == null || header.equals("")) {
        return "";
    }
    //得到用户的操作系统
    if (header.indexOf("NT 6.0") > 0) {
        systemInfo = "Windows Vista/Server 2008";
    } else if (header.indexOf("NT 5.2") > 0) {
        systemInfo = "Windows Server 2003";
    } else if (header.indexOf("NT 5.1") > 0) {
        systemInfo = "Windows XP";
    } else if (header.indexOf("NT 6.0") > 0) {
        systemInfo = "Windows Vista";
    } else if (header.indexOf("NT 6.1") > 0) {
        systemInfo = "Windows 7";
    } else if (header.indexOf("NT 6.2") > 0) {

```

```

        systemInfo = "Windows Slate";
    } else if (header.indexOf("NT 6.3") > 0) {
        systemInfo = "Windows 9";
    } else if (header.indexOf("NT 5") > 0) {
        systemInfo = "Windows 2000";
    } else if (header.indexOf("NT 4") > 0) {
        systemInfo = "Windows NT4";
    } else if (header.indexOf("Me") > 0) {
        systemInfo = "Windows Me";
    } else if (header.indexOf("98") > 0) {
        systemInfo = "Windows 98";
    } else if (header.indexOf("95") > 0) {
        systemInfo = "Windows 95";
    } else if (header.indexOf("Mac") > 0) {
        systemInfo = "Mac";
    } else if (header.indexOf("Unix") > 0) {
        systemInfo = "UNIX";
    } else if (header.indexOf("Linux") > 0) {
        systemInfo = "Linux";
    } else if (header.indexOf("SunOS") > 0) {
        systemInfo = "SunOS";
    }
    return systemInfo;
}

/**
 * 获取来访者的主机名称 * * @param ip
 * @return
 */
public static String getHostName(String ip) {
    InetAddress inet = null;
    try {
        inet = InetAddress.getByName(ip);
    } catch (UnknownHostException e) {
        logger.error("UnknownHostException", e);
    }
    return inet.getHostName();
}

/**
 * 执行参考
 */
public void transfer(HttpServletRequest request) {
    SystemUtils systemUtils = new SystemUtils();
    String ipaddr = systemUtils.getIpAddr(request);
    System.out.println(systemUtils.getIpAddr(request));
    System.out.println(systemUtils.getRequestBrowserInfo(request));
    System.out.println(systemUtils.getRequestSystemInfo(request));
    System.out.println(systemUtils.getHostName(ipaddr));
}
}

```