



链滴

Servlet 怎么样才能得到完整的 Http 请求输入流

作者: [zonghua](#)

原文链接: <https://ld246.com/article/1474598885679>

来源网站: [链滴](#)

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

<p>象这样只能得到请求体，不能获取到包括头部的整个二进制数据</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> @Override
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> protected void
ervice(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOExcepti
n {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     OutputStrea
Writer osw = new OutputStreamWriter(resp.getOutputStream());
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     InputStrea
s = req.getInputStream();
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     osw.write(Co
stant.TEST_WEBSOCKET_PAGE);
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     osw.flush();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }
</span> </span> </code> </pre>
```

<p>处理 Http 是 Web 容器的工作，AJP 将数据传给 Servlet 就已经是解析过的对象，因此不能直
获取到原本的内容</p>

<p>自己再拼接回去咯 :satisfied:</p>

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> GET http://www.ruanyifeng.com/static/themes/theme_scrapbook/theme_scrapbook.css H
TP/1.1
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Host: www.ruanyif
ng.com
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Connection: keep
alive
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> User-Agent: Mozil
a/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.27
5.116 Safari/537.36
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Accept: text/css,*/
;q=0.1
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Referer: http://w
w.ruanyifeng.com/blog/2016/08/http.html
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Accept-Encoding:
gzip, deflate, sdch
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Accept-Language:
zh-CN,zh;q=0.8
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> Cookie: _ga=GA1.
.1640937992.1474458063
</span> </span> </code> </pre>
```

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> public static final byte[] PROTOCOL_BYTES = " HTTP/1.1\r\n".getBytes();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> public static fina
byte[] CRLF_BYTES = "\r\n".getBytes();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> public static fina
byte BLANK_BYTE = ' ';
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> public static fina
byte[] COLON_BYTES = ":".getBytes();
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> public static fina
byte QUESTION_BYTE = '?';
</span> </span> <span class="highlight-line"> <span class="highlight-cl">
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> public byte[] ge
RequestRaw(HttpServletRequest req) throws IOException {
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     ByteArrayOu
putStream buffer = new ByteArrayOutputStream();
```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> // header val
es
</span></span><span class="highlight-line"><span class="highlight-cl"> bufferWriteB
tesFromStringLength(buffer, req.getMethod());
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.write(
LANK_BYTE);
</span></span><span class="highlight-line"><span class="highlight-cl"> bufferWriteB
tesFromStringLength(buffer, req.getRequestURL().toString());
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.write(
UESTION_BYTE);
</span></span><span class="highlight-line"><span class="highlight-cl"> bufferWriteB
tesFromStringLength(buffer, req.getQueryString());
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.write(
ROTOCOL_BYTES);
</span></span><span class="highlight-line"><span class="highlight-cl"> Enumeratio
<&String&gt; headerNames = req.getHeaderNames();
</span></span><span class="highlight-line"><span class="highlight-cl"> while (heade
Names.hasMoreElements()) {
</span></span><span class="highlight-line"><span class="highlight-cl"> String hea
erName = headerNames.nextElement();
</span></span><span class="highlight-line"><span class="highlight-cl"> String hea
erValue = req.getHeader(headerName);
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.writ
(headerName.getBytes());
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.writ
(COLON_BYTES);
</span></span><span class="highlight-line"><span class="highlight-cl"> bufferWrit
BytesFromStringLength(buffer, headerValue);
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.writ
(CRLF_BYTES);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.write(
RLF_BYTES);
</span></span><span class="highlight-line"><span class="highlight-cl"> // body value

</span></span><span class="highlight-line"><span class="highlight-cl"> InputStream
inputStream = req.getInputStream();
</span></span><span class="highlight-line"><span class="highlight-cl"> int nRead;
</span></span><span class="highlight-line"><span class="highlight-cl"> byte[] data =
new byte[1024];
</span></span><span class="highlight-line"><span class="highlight-cl"> while ((nRead
= inputStream.read(data, 0, data.length)) != -1) {
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.writ
(data, 0, nRead);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> buffer.flush();
</span></span><span class="highlight-line"><span class="highlight-cl"> byte[] byteAr
ray = buffer.toByteArray();
</span></span><span class="highlight-line"><span class="highlight-cl"> return byteAr
ray;
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> public String ge
RequestBase64(HttpServletRequest req) throws IOException {

```

```

</span></span><span class="highlight-line"><span class="highlight-cl"> byte[] rawByt
s = getRequestRaw(req);
</span></span><span class="highlight-line"><span class="highlight-cl"> return Base64
getEncoder().encodeToString(rawBytes);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> public String en
odeBase64BytesString(String base64Bytes) {
</span></span><span class="highlight-line"><span class="highlight-cl"> byte[] bytes
Base64.getDecoder().decode(base64Bytes);
</span></span><span class="highlight-line"><span class="highlight-cl"> StringBuffer s
ringBuffer = new StringBuffer();
</span></span><span class="highlight-line"><span class="highlight-cl"> for (byte b :
ytes) {
</span></span><span class="highlight-line"><span class="highlight-cl"> stringBuffe
.append((char) b);
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> return string
uffer.toString();
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl"> public void buff
rWriteBytesFromStringAllowNull(ByteArrayOutputStream byteArrayOutputStream, String stri
g) throws IOException {
</span></span><span class="highlight-line"><span class="highlight-cl"> if (string != n
ll) {
</span></span><span class="highlight-line"><span class="highlight-cl"> if (string.g
tBytes() != null) {
</span></span><span class="highlight-line"><span class="highlight-cl"> byteArr
yOutputStream.write(string.getBytes());
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span><span class="highlight-line"><span class="highlight-cl"> }
</span></span></code></pre>

```

那个 `ByteArrayOutputStream` 应该可以共用，需要吧所有的数据都给转到 `byte[]`

问一下那个 `bufferWriteBytesFromStringAllowNull` 应该是不是写成继承扩比较好，而不是调用自身

消息同步不会写，想想还是算了。。。