

JavaME应用设计指南之联网重定向 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_JavaME_E5_BA_94_E7_c104_144640.htm HTTP 1.1协议允许Web服务器临时改变资源的位置，也就是说你访问的资源在另外一个地址。

这时候服务器返回的响应代码是302，而新的地址存放在Header中，Header的名称是Location。正常情况下，客户端浏览器应该指向临时的访问地址。但是，移动终端设备差距很大，在处理302返回码的时候，设备之间的行为差异很大。

下面的代码可以用来处理重定向问题，但是在部分设备中会导致应用程序出错。 Connection c = (HttpConnection)

```
Connector.open(uri). int status = c.getResponseCode(). String  
new_uri = c.getHeaderField("Location"). // new_uri is null on some  
devicesif (status == 302) { c.close(). c = (HttpConnection)
```

```
Connector.open(new_uri). // Breaks here}
```

由于重定向是HTTP 1.1的特性，那么所有1.1兼容的设备都需要考虑这个问题。下面介绍如何解决这个问题。事实证明在某些设备上，底层的网络协议栈处理重定向的问题，302响应码告诉应用程序内部的处理流程。应用程序应该等待直到响应码等于302。但是有些设备不能正确地从响应中解析出Location字段，这样Location字段的内容是null，响应码存储在了响应的内容之中。有经验的工程师会采用下面的解决办法。1) 解析响应，在Location或者响应的内容中查找新地址，如果找到的话关闭以前的连接，转向新的连接。2) 如果什么也没有找到的话，那么等待10-1000ms，直到状态码从302转变为200。马上处理响应，当作没有错误发生。下面的代码能够很好的解决重定

向的问题，供大家参考和完善。

```
Connection c =
(HttpConnection) Connector.open(uri). int status =
c.getResponseCode(). String redirection =
httpConnection.getHeaderField("Location").if (status ==
HttpConnection.HTTP_TEMP_REDIRECT) { if (redirection !=
null) { // This the standard HTTP 1.1 behaviour, move on to the
redirection uri (basically restarting again). } else { // Parse the content
of the HTTP response, if any. // Lookup for a "Location" header, if
found, set value to the redirection variable if (redirection != null) { //
Since location was found, fall back to the standard behaviour. } else {
long begin_wait = System.currentTimeMillis(). while
(System.currentTimeMillis() - begin_wait < 100). response =
httpConnection.getResponseCode(). }. if (response == 200) { //
Once again were back on tracks, continue processing as if no error
has ever happen } else { // Here were really hopeless. Either the server
did provided a valid redirection uri, // or the device did not
preserved it. The best option is probably to fail by throwing an
exception. }. }. }.} else // Handle other error codes here} // Handle
success here (status == 200)
```

您还可以了解一下Http协议的细节，<http://www.ietf.org/rfc/rfc2616.txt>。本文是笔者在阅读SUN的技术文章的时候编译的。您可以通过下面的地址阅读原文，也欢迎您编译其他的好文章，共同促进国内Java ME技术的发展。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com