

cookie技术在JavaME平台的应用与实现 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/222/2021_2022_cookie_E6_8A_80_E6_c104_222331.htm Cookie在Web应用程序中被广泛采用，维护浏览器和服务端之间的状态。遗憾的是这一特性在Java ME平台中并没有得到支持。因此，要想维持客户端和服务端的状态则必须使用URL重写的方式。URL重写操作起来比较麻烦，所以研究一下cookie的原理并在Java ME平台上实现cookie是不错的尝试。首先，我们来看一下cookie的原理。当服务器需要和浏览器维持某一状态的时候，例如需要记录用户的购物车中已经购买的商品。这时候服务器可以新建一个Cookie并把它写入到响应中，浏览器从响应中接收到cookie并保存起来。当浏览器再次向服务器发送请求的时候，浏览器会根据域（domain）和路径（path）检查是否有匹配的cookie，如果有则把cookie以“名称=值”的形式发送给服务器，服务器从请求中解析出cookie就知道用户的状态了。那么，浏览器根据什么规则来决定向服务器发送cookie呢，首先要匹配domain，如果cookie的域属性是.google.com，那么请求指向j2medev.com的时候，cookie就不会被发送。如果域匹配的条件满足，则判断path是否匹配，如果cookie的path属性是请求的uri的父目录的话，那么cookie就会被发送给服务器。Cookie是有存活周期的，到期的cookie会被浏览器自动清除。如果服务器创建cookie的时候不设置生命周期，那么在会话结束后浏览器就会删除cookie。如果不为cookie指定path属性，那么默认就是这次请求的路径。cookie在很多web应用程序中都有应用，比如记住密码，购物车等。在开发MIDlet的时

候，你也可以让你的应用程序支持cookie，这样维持客户端与服务器端的状态将变得简单，为你集中精力解决其他业务方法奠定了基础。既然已经知道了cookie的工作原理，那么就on应该考虑一下在Java ME平台如何实现cookie，这个想法是否可行。我将从下面三个方面进行分析。第一：获得cookie当服务器端的响应到来的时候，我们应该能够读取cookie。如果服务器向客户端写入Cookie的时候，响应中的HTTP头

“Set-Cookie”中会包含一个字符串，代表了cookie的信息。幸运的是我们是用HttpConnection.getHeaderField("Set-Cookie")方法即可获得cookie，但是需要注意这里只是读取了一个cookie，如果响应中包含了多个cookie，那么你需要循环读取。类似于下面的代码

```
String sCookie = null. String key = null.
int i = 0. //如果key存在，则查询header的key，如果key等于SET_COOKIE，则存储 while((key =
connection.getHeaderFieldKey(i))!=null){
if(key.equals(SET_COOKIE)||key.equals(SESSIONID)){ sCookie
= connection.getHeaderField(i). saveCookie(sCookie,url). } i . } 上
面的代码把header是Set-Cookie和SessionID的cookie内容读取
下来。 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
```