**<https://msdn.microsoft.com/zh-cn/library/aa479314.aspx>**

当用户首次连接到站点时，将以内存块的形式创建一个全新的会话状态以存放数据，同时，还会创建一个 ID 以便将其与当前用户唯一地联系起来。当下一次发出请求时，该用户将被要求提交该会话 ID，以便检索并正确地还原会话状态。会话 ID 是 ASP 和 ASP.NET 完全自主生成的字母数字字符串。用户如何管理它并确保用户每个后续请求来包装它呢？

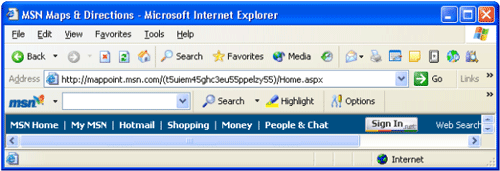
Cookie 存储会话的 ID，

当启用了 Cookie 的浏览器收到响应数据包时，它将寻找附加的 Cookie，并将它们的内容存储到本地 Windows 目录中特定文件夹的某个文本文件中。Cookie 还包含有关该源站点的信息。接下来，当浏览器向该站点发送请求时，它会在 Cookie 文件夹中查找源自该域的 Cookie。如果找到，则该 Cookie 自动附加到传出的数据包中。该 Cookie 将命中服务器应用程序，并在此被检测、提取和处理。

在 ASP.NET 中，无需使用 Cookie，就可以有选择地建立必要的会话-用户联系。非常有趣的是，除了以下配置设置以外，您无需在 ASP.NET 应用程序中更改任何内容即可启用无 Cookie 会话。

<sessionState cookieless="true" />

当不使用 Cookie 时，ASP.NET 在哪里存储会话 ID 呢？在这种情况下，会话 ID 插入到 URL 内的特定位置中。下图显示一个使用无 Cookie 会话的真实站点的快照。



http://yourserver/folder/(session ID here)/default.aspx

会话 ID 嵌入到 URL 中，并且无需在其他任何地方持久保存它

实现方式

当新浏览器会话的第一个请求进入时，会话状态模块读取 web.config 文件中有关 Cookie 支持的设置。如果 节的 **cookieless** 属性设置为 true，则该模块生成一个新的会话 ID，通过将该会话 ID 填充到资源名称前面的相邻位置来分割 URL，并且使用 HTTP 302 命令将浏览器重定向到新的 URL。

当每个请求到达 IIS 入口时（远远早于它被移交给 ASP.NET），aspnet\_filter.dll 获得了一个查看它的机会。如果该 URL 将会话 ID 嵌入到括号中，则会提取该会话 ID 并将其复制到一个名为 **AspFilterSessionId** 的请求标头中。然后，重写该 URL 以使其看起来像原来请求的资源，并且将其释放。这一次，ASP.NET 会话状态模块从请求标头中检索会话 ID，并且通过会话-状态绑定继续工作。

只要该 URL 包含可用来获取会话 ID 的信息，无 Cookie 机制就可以很好地工作。正如您稍后将看到的那样，这会造成一些使用限制。