禁用cookie时咋办

1.禁用cookie的客户会忽略响应的Set-Cookie首部信息

2.此时**URL重写**是一条后路（当然HTML5的localStorage也是可以的，这里先不说）

我们的目的是让用户和容器交换会话ID信息，来回传cookie是最简单的，如果不能把ID等状态信息放在cookie中，那么把ID放在URL中也是可以的。

3.在响应发回的HTML中会把会话ID追加到页面中所有URL的最后，前面用特定符号隔开

在请求中，把会话ID放到URL后

4.cookie不能工作时，容器就会求助URL重写，但是程序员需要对响应中发送的所有URL完成编码，此时URL重写才奏效；

response.encodeURL()

如果禁用了cookie的同时，不对URL进行编码，会话就会失效；

容器只有使用cookie失效才会用URL重写。

5.因此，当调用request.getSession，且容器知道要建立新的会话的时候，容器会做两件事，一是针对会话ID有一个Set-Cookie首部，二是会向URL追加会话ID（假设使用了response.encodeURL()）

6.还有一种可能：想重定向URL，但还想使用一个会话，可用以下URL编码方法

response.encodeRedirectURL(“/Beer.do”)

7.使用URL重写只有一种可能：作为会话一部分的所有页面都是**动态**生成的

因此，可以在JSP中完成，还有个JSTL标签可用<c:URL>

8.URL重写是自动的！只有当你对URL完成了编码时才奏效~（会对html中所有的URL也进行重写）encodeURL和encodeRedirectURL

9.URL编码只与响应有关！由response调用