

表单重复提交问题的解决方法 PDF转换可能丢失图片或格式
，建议阅读原文

https://www.100test.com/kao_ti2020/468/2021_2022__E8_A1_A8_E5_8D_95_E9_87_8D_E5_c67_468744.htm (1) 在用户进入表单页面的时候生成一个随机的数字，并赋值给一个session,并写入一个隐藏控件 在处理提交数据的页面判断传入的隐藏控件的值是否=session的值。相等表明允许提交 最重要的一步：处理提交的数据成功后清空session，这样他反复提交过来的页面，隐藏控件的值是存在的，而session为空，就backup页面，告诉他，不能提交了 (2) 用Struts的Token机制解决表单重复提交 Struts 的Token（令牌）机制能够很好的解决表单重复提交的问题，基本原理是：服务器端在处理到达的请求之前，会将请求中包含的令牌值与保存在当前用户会话中的令牌值进行比较，看是否匹配。在处理完该请求后，且在答复发送给客户端之前，将会产生一个新的令牌，该令牌除传给客户端以外，也会将用户会话中保存的旧的令牌进行替换。这样如果用户回退到刚才的提交页面并再次提交的话，客户端传过来的令牌就和服务器端的令牌不一致，从而有效地防止了重复提交的发生。这时其实也就是两点，第一：你需要在请求中有这个令牌值，请求中的令牌值如何保存，其实就和我们平时在页面中保存一些信息是一样的，通过隐藏字段来保存，保存的形式如：
`input type="hidden" name="org.apache.struts.taglib.html.TOKEN" value="6aa35341f25184fd996c4c918255c3ae"`，这个value是TokenProcessor类中的generateToken()获得的，是根据当前用户的session id和当前时间的long值来计算的。第二：在客户

端提交后，我们要根据判断在请求中包含的值是否和服务器的令牌一致，因为服务器每次提交都会生成新的Token，所以，如果是重复提交，客户端的Token值和服务器端的Token值就会不一致。下面就以在数据库中插入一条数据来说明如何防止重复提交。在Action中的add方法中，我们需要将Token值明确的要求保存在页面中，只需增加一条语句

```
: saveToken(request)., 如下所示： public ActionForward  
add(ActionMapping mapping, ActionForm form,  
HttpServletRequest request, HttpServletResponse response) //前面的  
处理省略 saveToken(request). return  
mapping.findForward("add"). }在Action的insert方法中，我们根据  
表单中的Token值与服务器端的Token值比较，如下所示：  
public ActionForward insert(ActionMapping mapping, ActionForm  
form, HttpServletRequest request, HttpServletResponse response) if  
(isTokenValid(request, true)) ...{ // 表单不是重复提交 //这里是  
保存数据的代码 } else ...{ //表单重复提交 saveToken(request). //  
其它的处理代码 }} 其实使用起来很简单，举个最简单、最需要  
使用这个的例子：一般控制重复提交主要是用在对数据库  
操作的控制上，比如插入、更新、删除等，由于更新、删除  
一般都是通过id来操作（例如：0updateXXXById,  
removeXXXById），所以这类操作控制的意义不是很大（不  
排除个别现象），重复提交的控制也就主要是在插入时的控  
制了。 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
```