**Mutli-click prevention using struts tokens- Prevent Duplicate Submission**   
saveToken() - generate the token key and save to request/session attribute.   
isTokenValid() - validate submitted token key against the 1 store in request/session.  
resetToken() - reset the token key  
Follow the steps to setup Mutli-click prevention using struts tokens.   
  
**Step 1. Action Class where saveToken() before JSP Page.**   
  
First saveToken() then forward to your jsp.   
Upon loading the form, invokes saveToken() on the action class to create and store the token key.   
Struts will store the generated key in request/session.  
  
  
public class LoadAction extends Action   
{  
public ActionForward execute(ActionMapping mapping,ActionForm form,  
HttpServletRequest request,HttpServletResponse response)  
{ ActionForward forward;  
saveToken(request);  
forward=mapping.findForward("empformpage");// this is the jsp page where you want to struts tokens.  
return forward;  
}  
}  
  
  
**Step 2. Store Token key as a hidden field: ( empform.jsp)**   
  
In the browser if you type :   
<http://localhost:8080/testApp/loadActio.do>   
This will call execute() method of LoadAction.   
Then saveToken(request);//create and store the token key   
and mapping.findForward("empformpage"); forward to empform.jsp (below code is for empform.jsp)   
  
<%@ page import="org.apache.struts.action.Action"%>  
<%@ page import="org.apache.struts.taglib.html.Constants"%>  
<%@ taglib uri="/WEB-INF/struts-tiles.tld" prefix="tiles" %>  
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>  
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>  
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>  
<html:form name="MyForm" method="post" action="/submitAction.do">  
<html:text property="firstName" size="20" maxlength="50"/>  
<input type="hidden" name="<%= Constants.TOKEN\_KEY %>"  
value="<%= session.getAttribute(Action.TRANSACTION\_TOKEN\_KEY) %>" >  
<html:submit >Save</html:submit>  
</html:form>  
  
  
**Step 3. Check Token is Valid ?**   
  
Once the form submitted, invokes isTokenValid() on the action class,   
it will validate the submitted token key(hidden field) with the token key stored previously on request/session.   
If match, it will return true.  
  
public class SubmitAction extends Action  
{  
public ActionForward execute(ActionMapping mapping ,ActionForm form ,HttpServletRequest request,  
HttpServletResponse response)  
{  
EmpForm frm=(EmpForm)form;  
if(isTokenValid(request))  
{ // This is Not Duplicate Submission of the form  
// You can add your logic here   
resetToken(request);  
return mapping.findForward("sucess");  
}  
else  
{ // This is Duplicate Submission of the form  
// Return to the JSP to display the error message ( This is Duplicate Submission);  
return mapping.findForward("duplicatesubmitpage");  
}  
}  
}