How to handle HTTP GET/POST in a Portlet JSP?

Introduction

When implementing a FORM in a normal JSP, we set the action attribute of the FORM tag with the relative path of servlet/JSP that is going to receive the GET/POST request. We also set the name attribute of the INPUT tag to a unique name.

However, when implementing a FORM in a portlet JSP, we do not know during development time the relative path of our portlet's JSP. We also need to ensure that the input field names will not clash with the input fields of other portlets in the portal page. This tip list the steps how we can get around these two problems.

Steps

Creating the portlet's URL for the FORM's action attribute

1. In the portlet's doView()/doEdit() method, call response.createURI() to create the URL for the action attribute. The same method (doView(), doEdit() etc) will be used to handle the GET/POST request.
   eg `PortletURI s1URI = **response.createURI()**`
2. Store the returned value (as a String) in the portlet's request or java bean.
   eg `request.setAttribute("actionVal", **s1URI.toString()**);`
3. In the JSP page, retrieve the String from the portlet request or bean.
   eg `<jsp:useBean id="**actionVal**" class="java.lang.String" scope="request" />`
4. Set the retrieved String as the value for the action attribute of the FORM.
   `<FORM method='POST' action='**<%=actionVal%>**'>`

Creating a unique name for the input element

1. In the JSP, use the portletAPI tag library to create a unique name for your input field
   eg `<input type="text" name='**<portletAPI:encodeNamespace value="firstpage"/>**'>`
   You can also call the encodeNamespace() method of implicit portlet response object to achieve the same result.
   eg `<input type="text" name='**<%=response.encodeNamespace("firstpage")%>**'>`

Receiving the GET/POST request

1. In your doView() or doEdit() method, retrieve the parameter value normally, using the unencoded input field's name as the parameter string.
   eg `String firstpage = **request.getParameter("firstpage");**`