YmtdSampleApplication

Sample Application

It is the belief of the Velocity developers that you should not have to specially code your applications to work around issues that are related directly to Java.

In other words, one of the strong arguments of the JSP/Struts community is to say something to the effect of: "this is a poor example of using JSP." While this may be true, the fact of the matter is that nearly every example available really is a poor example of using JSP. This goes back to the statement that says that embedding Java code in your page is a bad thing. Yes, we all know that now.

If one reads the articles available on JavaWorld that are about JSP, nearly every single article has one thing or another in it that demonstrates poor usage of the tool. Why is it that so many (obviously) talented people cannot come up with correct examples of using the tool?

The truth is that it is very hard to use the tool correctly. Struts is doing an excellent job of making it easier and attempting to show the right way, however it is simply hiding the uglyness of the original design of JSP.

Object Oriented design dictates that you extend a class to add functionality to the base class. The publicly available methods in the base class are still available to the classes that extend it. Putting Struts on top of JSP doesn't fix the warts in JSP. It simply hides them until your developers find them.

```
<%-- toolview.jsp --%>

<%
  String title = "Tool Listing";
  String deck = "A list of content creation tools";
  String desc = "Without tools, people are nothing more than animals.";
%>

<%@ include file="/header.jsp" %>

<%@ page session="false" %>
<%@ page errorPage="/errorTaker.jsp" %>

<jsp:useBean id="toolbean" class="ToolBean" scope="application">
  <jsp:setProperty name="toolbean" property="toolsFile"
                value='<%= application.getInitParameter("toolsFile") %>' />
</jsp:useBean>

<%
  Tool[] tools = toolbean.getTools(request.getParameter("state"));

  for (int i = 0; i < tools.length; i++) {
    Tool tool = tools[i];
%>
  <HR SIZE=2 ALIGN=LEFT>

  <H3>
  <%= tool.name %>

  <% if (tool.isNewWithin(45)) { %>
    <FONT COLOR="#FF0000"><B> (New!) </B></FONT>
  <% } else if (tool.isUpdatedWithin(45)) { %>
    <FONT COLOR="#FF0000"><B> (Updated!) </B></FONT>
  <% } %>

  </H3>
  <A HREF="<%= tool.homeURL %>"><%= tool.homeURL %></A><BR>

  <%= tool.comments %>

<% } %>

<%@ include file="/footer.jsp" %>
```

Because of JSP whitespace preservation rules you must be careful when writing if/else statements with scriptlets. The following code would *not* work:

```
  <% if (tool.isNewWithin(45)) { %>
    <FONT COLOR=#FF0000><B> (New!) </B></FONT>
  <% } %>
  <% else if (tool.isUpdatedWithin(45)) { %>
    <FONT COLOR=#FF0000><B> (Updated!) </B></FONT>
  <% } %>
```

With this code the background servlet would attempt to print a new line between the if and else clauses, causing the obscure compile error: 'else' without 'if'.

This is the version of the example above translated to Velocity:

```
## toolview.vm

#set ($title = "Tool Listing")
#set ($deck = "A list of content creation tools")
#set ($desc = "Without tools, people are nothing more than animals." )

#parse ("header.vm")

$toolbean.setToolsFile($application.getInitParameter("toolsFile"))

#set ($tools = $toolbean.getTools($request.getParameter("state")))

#foreach ($tool in $tools)
  <HR SIZE=2 ALIGN=LEFT>

  <H3>
  $tool.Name

  #if ($tool.isNewWithin(45))
    <FONT COLOR="#FF0000"><B> (New!) </B></FONT>
  #elseif (tool.isUpdatedWithin(45))
    <FONT COLOR="#FF0000"><B> (Updated!) </B></FONT>
  #end
  </H3>
  <A HREF="$tool.homeURL">$tool.homeURL</A><BR>

  $tool.comments
#end

#parse ("footer.vm")
```

You make the decision.

YmtdJavaBeans <- Previous | Next -> YmtdTaglibs