

# JspVersusVelocity

## Case Study: JSP vs. Velocity

I just wanted to share my findings on Velocity. I use velocity as an embedded mail templating system in our application server.

When I began this subproject, I was using JSP as the language for generating template output for our application server. It was horrible. There were many problems, not to mention the difficulty of embedding a servlet engine in the app server that I could communicate with through direct method calls instead of over a socket. Specific to the Velocity argument was the JSP template versus the VTL template. I'd like to show you what my JSP template looked like.

Package names have been changed to protect the innocent. Here is an example:

```
<%@ page session="false"
    import="java.util.*,
    foo.events.*,
    foo.emmeeting.*,
    foo.validation.*,
    foo.util.actionevent.ActionEvent,
    java.util.Date"%>

<jsp:useBean id="meeting" scope="request" type="foo.IMeeting"
class="foo.Meeting"/>

<html>

<!-- etc -->

<table width="600" border="0" cellspacing="0" cellpadding="4"
    bgcolor="white">
<tr>
<td>
<%
Iterator          events = meeting.getStoredEventsIterator();
int               repId = 0;

%>
<i><b>Topic: <%= meeting.getTopic()%></b></i>
<p>
<%
while (events.hasNext()) {
    Object event = events.next();
    Participation yapper =
meeting.getParticipation(((MeetingEvent)event).getFromId());
    if(event instanceof ChatEvent) {
        if("R".equals(yapper.getRole()) ) {
%>
<font color="#000000"><b><%=yapper.getName()%></b></font>
<%
        repId = yapper.getParticipantId();
        } else {
%>
<font color="#0000f0"><b><%=yapper.getName()%></b></font>
<%
        }
        ChatEvent ce = (ChatEvent) event;
%>
<br>
<%
        } else if(event instanceof URLPushedEvent) {
            if("R".equals(yapper.getRole()) ) {
%>
<font color="#000000"><b><%=yapper.getName()%></b></font>
<%
            } else {
%>
<font color="#0000f0"><b><%=yapper.getName()%></b></font>
<%
            }
            URLPushedEvent pe = (URLPushedEvent) event;
%>
```

```

<a href="<%=pe.getStoredData()%>">"<%=pe.getStoredData()%>"</a> <br>
<%
}
}
%>
<hr>
<br><%=meeting.getMeetingId()%>:<%=repId%><br>
</td>
</tr>
</table>
</div>
</body>
</html>

```

Seasick yet? Now here is the Velocity template accomplishing the same thing as above:

```

<html>

<!-- etc -->

<table width="600" border="0" cellspacing="0" cellpadding="4"
  bgcolor="white">
<tr>
<td>
<i><b>Topic: $meeting.getTopic()/b></i><p>

#foreach( $event in $meeting.getStoredEventsIterator() )
  #set( $yapper = $meeting.getParticipation($event.getFromId()) )
  #if( $event.getClass().getName().equals($urlEvent) )
    #if( $repRole.equals($yapper.getRole()) )
      #set( $repId = $yapper.getParticipantId() )
      <font color="$blk"><b>$yapper.getName()/b></font>
    #else
      <font color="$blu"><b>$yapper.getName()/b></font>
    #end
    <a href="$event.getStoredData()">$event.getStoredData()/a> <br>
  #elseif( $event.getClass().getName().equals($chatEvent) )
    #if( $repRole.equals($yapper.getRole()) )
      #set( $repId = $yapper.getParticipantId() )
      <font color="$blk"><b>$yapper.getName()/b></font>
    #else
      <font color="$blu"><b>$yapper.getName()/b></font>
    #end
    $event.getStoredData() <br>
  #end
#end
<hr>
<br>$meeting.getMeetingId():$repId<br>
</td>
</tr>
</table>
</div>
</body>
</html>

```

I was subsequently able to define that largish foreach loop into a macro and shrink it even more, using Velocity's #macro function, so I could do this:

```
#drawStandardHtmlTranscript ( $meeting )
```

removing 20 lines of code from each template whilst only adding 2 lines (one for #parse and the other for the

```
#drawStandardHtmlTranscript.
```

The call to the Velocity runtime to finally merge the template consumes about 20-50 milliseconds (on a 400 mHz PII Linux system) when I pool the contexts (time doubles when I do not). This is great performance, too!

Last but not least, teaching the web design team to write a template took literally 10 minutes from start to finish. They haven't asked me a single question since.

---

Nick Bauman

Software Engineer

3600 Dupont

Minneapolis, MN

55412