

Re: Recommendations for a web application framework?

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2006-07/msg00709.html>

- *From:* Timo Stamm <timo.stamm@xxxxxxxx>
 - *Date:* Sun, 09 Jul 2006 00:21:59 +0200
-

Chris Smith schrieb:

Timo Stamm <timo.stamm@xxxxxxxx> wrote:

In a different form, it's also the Wicket example you posted, which has two id attributes in the page corresponding to the two components; and then a Java class representing that page, which again has to match the id attributes on components on the page. It's a little better than Struts, to be sure. Yet I still see this duality wherein there's a Java class representing the form.

Unfortunately, you can not eliminate this duality if you want to use HTML templates and insert a value into a specific position in the template.

I think it is avoidable, and I think JSF does it.

To be perhaps a little clearer, what JSF does is ensure that the correspondence between identifiers and names in Java code and HTML content is one-way. That eliminates the really painful bit, which is to try and keep stuff in sync.

Yes, JSF avoids the duality because it doesn't really have templates. You don't have to bind a component to a position in the template, you define components directly in the JSP code using taglibs, and this introduces problems:

- It is impossible to use HTML mockups from a web designer without major changes that render them useless to the web designer.
- Browser incompatibilities are annoying. Having to debug them on a running application instead of the actual HTML file makes the issue worse.
- It forces you to express logic in a XML language. Java may have a verbose syntax, but it's still much better than XML, especially with a decent IDE.

Re: Recommendations for a web application framework?

I can actually write a JSF backing bean that encapsulates what I need to know in order to perform some task... and I really don't care whether the user interface