

Professional JSP

Struts

Very brief overview of MVC and web application frameworks. Point to other resources for more detailed explanations.

Explain difference between traditional MVC vs. webapp MVC as described at <http://lists.ourshack.com/pipermail/templates/2002-November/003974.html>

Explain that the purpose of the chapter is to highlight advanced features of Struts, not to evangelize struts or compare it to other frameworks.

Explain that this chapter will cover the following Struts topics: IDE Tools (i.e. Struts Console, Easy Struts), Validator, Tiles, Exceptions, Modules, DispatchAction, DynaFormBeans, XDoclet, Struts Tags (including JSTL and JSP 2.0 syntax) + XHTML and CSS.

Point out that there are other MVC Framework alternatives (i.e. Maverick, Webwork), and to always look for an active development/user community when choosing your framework. I'm not saying that these frameworks don't have one. How does JSP 2.0 and Servlets 2.4 affect Struts-based webapps? It doesn't much, it just makes for less typing in your JSPs.

Code-generation tools, such as XDoclet, take much of the pain out of writing XML files (i.e. web.xml, struts-config.xml, validation.xml). Mention value of research into these tools vs. digging and and refactoring later.

Relate personal experience using Struts – headaches and hoorays!

Advanced Features

Explain that most books/articles seem to cover what Struts is rather than the good stuff.

Describe that this chapter refers to Struts 1.1, and a nightly build (because it has XHTML support) rather than 1.1 beta 2.

Explain stability of Struts (even the betas) and by using XDoclet you can easily test new versions.

Mention importance of flexible directory structure and give recommendation (**screenshot**).

Struts Tools

Struts Console

Easy Struts

XDoclet. Building ValidatorForms, JSPs and .properties file using

XDoclet. **Code sample**

Ant **sample build file**

More??

Validator

Compare to ActionForm with validate() method. **Code example**

Client side vs. Server Side

Building using XDoclet **Code example**

Tiles

Explain using Tiles vs. JSP includes: **Code example**

Using a definition in a JSP, vs. in tiles-config.xml.

Benefits/drawbacks.

How to use CSS and/or tiles to customize a UI for different users.

Exceptions

App/Module Exceptions and using execute method vs. perform
Chaining Exceptions

Rendering messages to the UI for all exceptions, user should never see an error page.

Exception Chaining and how to use.

Mention it's easy to render both errors and messages, so success messages are also important.

Modules (was Sub-Applications)

Why should you use them? Explain benefits of different (large) features might call for different modules.

How to use them? SwitchAction. Should probably provide code sample.

Mention support only exists for extension mapping (i.e. *.do vs. /do/*).

Relate experience of developing sample sub app and recommend/don't recommend based on experience.

Built In Actions

Explain DispatchAction, LookupDispatchAction, ForwardAction, IncludeAction and SwitchAction. Provide scenarios of when to use.

DynaFormBeans

Explain that they were created to minimize the need for coding forms. Provide example declaration in a struts-config.xml

Provide **code examples** of:

getting/setting attributes of a DynaForm in an Action class

creating a DynaForm in an Action for use in a form

Tag Libraries

Brief explanation of each: html, bean, logic, nested, tiles

Now supports XHTML Transitional.

Describe how JSTL and JSP 2.0 Syntax can replace most logic and bean tags, but sometimes the struts tags require less typing. Also mention new Struts EL tags.

Briefly describe nested tag library. **Code example of list with children that can be added/deleted/modified within the list view.**

Explain web.xml syntax and taglibs declaration portability across app servers and what I recommend. Also, notify that some servers can read from .jar file and no tld needed.

Recommend Display Tag Library for paging/sorting lists of data.

Explain difficulties with exporting and using Tiles.

Struts Menu Tag Library.

Persistence in Struts

Explain how there are many options, but none seem to be recommended as a best practice by the Struts Development Team.

Explain that the datasource attribute in struts-config.xml uses DBCP under the covers (same as Tomcat), so you've got a built-in connection pool without extra work.

Relate experience of using patterns (ServiceLocator, BusinessDelegate, DAO) and how sometimes it seems it'd be easier to just call DAO's from Actions. Explain that being able to run tests against Business Delegates was a nice benefit.

Relate opinion of "just get it done" versus "doing it the right way" and that it really depends on the client/project and your level of expertise.

Show **code samples** of calling a SessionBean, BusinessDelegate, and a DAO from an Action.

Mention that EJBs have gotten a bad rap, but most likely because of developers doing them for their resume, and that XDoclet makes it much easier to develop EJBs.

Plan for extensibility/pluggability, but realize that "getting it done" might be the key factor to developing the app more so than "doing it right."

If time permits, provide a **code sample** of calling different persistence mechanisms. Even better, build these mechanisms using XDoclet from POJO's (Plain Old Java Objects). Better yet, allow for switching of persistence mechanisms at build time. Consider the following: DAOs, Castor/Hibernate, CMP Entity beans. Use Middlegen if you can.

Testing Struts

StrutsTestCase and Cactus make testing your Actions much easier.

Run your tests using Ant, and include/exclude cactus jars/configuration made simple.

Build JUnit tests for your BusinessDelegates and Persistence Layer.

Use JUnitDoclet with XDoclet to create shells of tests for your methods.

Mention ConfigurableTestCase and JUnit templates for IDEs.

Explain JSP Testing options: HttpUnit, JsUnit

(<http://www.edwardh.com/jsunit/>) and how compiling your JSPs before deploying will have you man headaches.

Use validators from the w3c to validate your HTML and CSS! It'll make your app much easier to test on multiple browsers/platforms. Ditch Netscape 4.x and other non-compliant browsers if you can.

Explain that testing might be a pain, but it's the shortest path to quality software and a stable product.

Tips and Tricks

Using Tiles' definitions as forwards rather than JSPs.

Refer to Security chapter on form-based authentication and roles.

Use Cookies to remember user preferences, stylesheets (mention stylesheet switcher).

Use CSS – it'll greatly reduce the size of your pages, and it's just plain easier.

Others??

Summary

Don't re-invent the wheel, someone has probably done what you are trying to do. Spend the time to research your problem and solutions.

Get involved with the technology you're using. Subscribe to the struts-user mailing list and watch your Inbox fill up. Search the archives first! Use XDoclet or other code-generation tools to make your initial development easier. Look for ones with a large community – that way, it'll fix itself.

Anything that is possible in a normal JSP/Servlet application is possible when using Struts, it just makes it easier. Likewise, anything that is possible on an HTML page is possible in a JSP.

You're on the bleeding edge, enjoy it!

Tell why I use Struts and why I will continue to use Struts.

Encourage use of sample app (struts-XDoclet) to jump-start new Struts app development.

Don't be afraid to refactor an existing app – the hard part is convincing the customer to do it, not actually doing it. If you don't have tests written, write them first, and then test that the same functionality exists after refactoring.

Describe Struts Roadmap and how JSRs always seem to have Struts in mind.

Talk about Java Server Faces and how Struts Tools/XDoclet will make developing webapps even easier. Mention .NET and advise developers to learn the specs and XML files so they don't just become an IDE experts.

Open Issues

What are the hardest concepts for Struts Newbies to understand?

Can I reference other books (i.e. Struts in Action, Java Development with Ant, JSTL in Action) for further details.

I need to query the Struts Development team and Users for recommended tools, not just ones they've heard of.