Java-Struts 2

Course Overview

Course Aims

The course aims to give participants an introduction to

- Java Programming Language
- the use of the Struts Framework to build simple Java-Web Applications.
- The 'Structured workshop' format is intended to allow delegates to construct a Web- project relevant to their immediate work needs. Formal slides will be presented to give supporting information where appropriate.

It is expected that the course will change in response to delegates experience, progress with the practical examples and professional needs.

Tools used will include Java 5, Apache Tomcat and the Eclipse IDE. All of these tools are embedded in the WebSphere Application Developer tool (WSAD)

Covered on this course

- Installation of a Java 5, Eclipse (WSAD) and Tomcat
- Use of Eclipse (WSAD) to build and run Java Programs
- Java Language Basics: Syntax , Javadoc , methods, variables, loops and conditionals.
- Just enough OO (Object Orientated) to get by.
- Basic Exception Handling
- Usage and where to get more information on the standard set of Libraries provided with Java. Where to find more specialised libraries to meet specific needs.
- Developing Web applications
- Struts – building , configuring and deploying web applications
- Further resources and references

Not covered on the course

The course aims to cover a wide range of material (both introduction and advanced) within 2 days. Unless delegate progress is very rapid , it is unlikely that the following topics (as an example) will be covered:

- Deployment and build of web archives (war) files or build tools such as ANT. Configuration of Web / Application servers not covered.
- Database access mechanisms and scalable 3 tier programming (including business rules and workflow).
- Enterprise Java , other than needed for a basic web application.
- Advanced Java Techniques such as Design and Architecture, Threading and File Handling and Scalability considerations.
- Java project quality and performance techniques.

These areas would be suitable for a follow up / refresher course within approx 12 weeks.

Pre-Requisites

- 1yr programming Experience (any language)
- Exposure to basic Web Programming (e.g. Awareness of HTML).
- Familiarity with an IDE (e.g. Visual Studio or Eclipse).
- Knowledge of Server side programming would be beneficial , but not required.

Facilities provided

- CD / Website with the above tools
- Knowledge base (Web based Wiki) providing notes and samples gathered during the 2 days.
- Internet Enabled PC's with supporting training facilities.

Downloads

Java 5 (JDK) :
http://java.sun.com/javase/downloads/index.jsp .
Take the link 'JDK 5.0 Update 10'. Use defaults in install program .

Tomcat 5.5 :
http://apache.mirrors.esat.net/tomcat/tomcat-5/v5.5.20/bin/apache-tomcat-5.5.20.zip

Unzip or install in C:\Software\Tomcat_5_5\

Eclipse 3.2:
http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.2.1-200609210945/eclipse-SDK-3.2.1-win32.zip

Unzip or install in C:\Software\Eclipse\

Struts 2 (Beta):
http://apache.oss-mirror.org/struts/binaries/struts-2.0.1-all.zip

Leave on Desktop – we install as part of course

Introduction to Java

Session 1 – Course Start

- Delegates Experience / what they want from course / practical example
- Introduction to presenter
- Course outline
  - ○Workshop format
  - ○Working in Pairs
  - ○Cover as much as we can
  - ○Wiki as a resource

- Housekeeping
  - ○ General IACT
  - ○ Timings

- *Slides – Java Resources*
  - Java
  - Javadocs
  - Eclipse
  - Apache Tomcat
  - Apache Struts 2

- *Practical:*
  - Download and Install Java 1.5
  - Download and Install Eclipse
  - Setup Eclipse Project

- *Demo*
  - How to test Version of Java installed
  - Run Eclipse and Tour of Eclipse features

Session 2 -First Program in Java

Syntax and the basics

- *Slides*
    - Selected slides on Java Syntax (Java Nuts and Bolts – slides 14-116)
        - Variables, Loops , Conditional , Arrays and output
- *Demo*
    - Simple Program using
        - Eclipse Wizard
        - public static void main
        - demo calling from command line and Eclipse
- *Practical*
    - Write a program to print hello world 100 times
        - for the last 50 loops, print 'goodbye world' instead of hello world
    - Run this from Eclipse and from the command line
    - (Optional) Modify this to take a parameter from the command line and print as part of the loop.

Session 3 – Using Methods

Also cover more advanced language features. (Java Nuts and Bolts – slides 14-116)

- *Slides*
  - Method visibility
  - Parameters
  - Operator Overloading
  - Return Types
- *Practical*
  - Write a program to guess a number between 1 and 100
- *Demo*
  - Simple method on how to read number from the command line.

Session 4 –Libraries

Get Delegates familiar with the features that come with Java Standard Edition , how to use , and where to find the documentation.

- *Slides*
  - Presentation (from Sun Website) of the common Sun Javadocs
- *Demo*
  - ○*Small Whiteboard session – the problems with System.out*
  - ○Logging
    - xml configuration files
- *Practical*
  - ○ Put the Strings (Hello / Goodbye) into a Java Collection
  - ○ Loop and read from the collection , then print.
  - ○ Convert the previous sample to use Log4j instead of System.out
  - ○ Log at the Debug Level in your code.
  - ○ Log the number of loops at the info level.
  - ○ Turn the logging
    - Off at the console, on at the Logfile
    - Debug to the Logfile , on at the console

Session 5- Classes And Packages

Just Enough OO to get By. Include a simple Inheritance and interfaces.

- *Slides*

  (Java Nuts and Bolts – slides 14-116)

  Statics

  Packages

- *Practical*

  1.JavaBean – write a javabean to hold the following values: Name , Address , Telephone number

  2.Put JavaBean into collection

  3.Loop and print them out

  4.Add a message when the constructor is called

  5.Add a variable to the JavaBean at class level and make it static. Create two instances of the same java bean. Show that a change to the static value on one javabean changes the static value on the other.

- *Demo*

  Working example of the above practical

Session 6 – Exception Handling

What happens when things go wrong

- *Slides*
  - High Level overview of Exceptions
  - At method level
  - Try … catch … finally
- *Demo*
  - If required – show the following (or let find from Javadoc)
  - Code snippets to
    - Read from a file
    - Write to a file
- *Practical*
  - Write a program to write information from a file
  - Write a 2nd program to read information from a file
  - Delete the file , and see what happens
  - Write exception handling code to cover the situation gracefully

Session (Optional) Java and Junit

- *Slides*
  - Why unit test?
- *Demo*
  - Create a sample method
  - Create a sample junit test
- *Practical*
  - Refactor

Session (Optional) More Advanced Java

- *Slides*
  - Optional – Oop2
  - Optional - Strings
- *Demo*
- *Practical*

Conclusion

- Summary of Day
- What to cover tomorrow

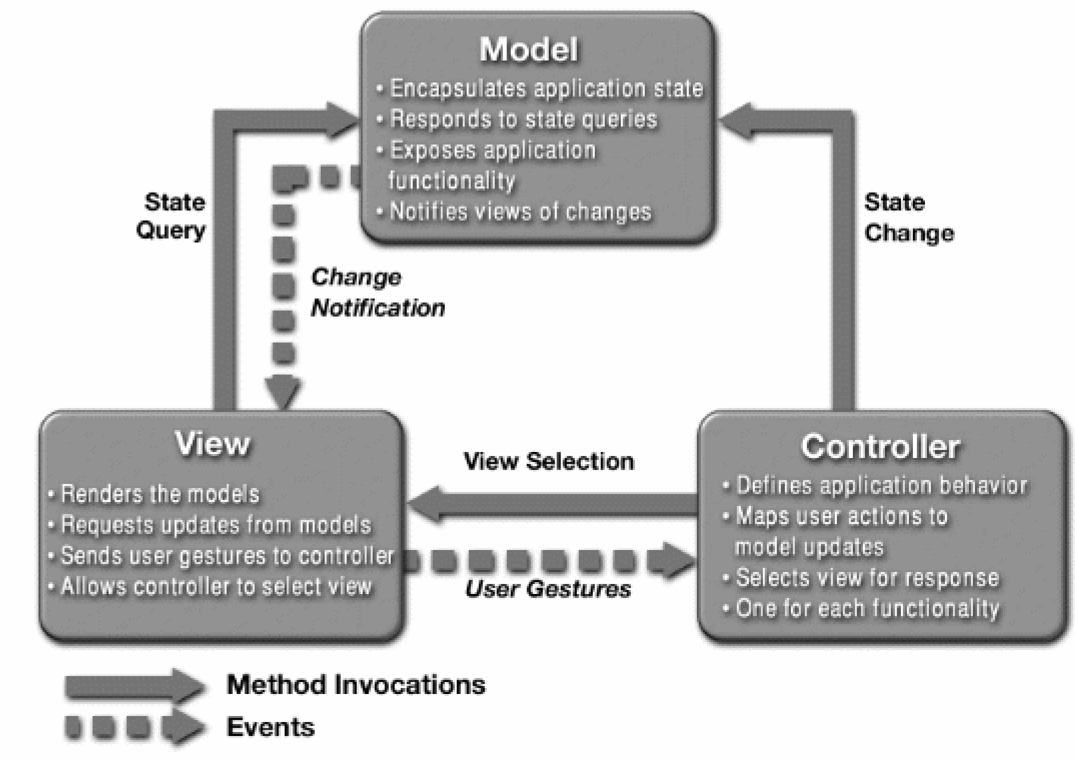
Introduction to Struts 2 and Java-Web Programming

Session 7 – Simple Model View Controller

- *Slides*
  - *What is MVC*
  - *Picture of Excel Spreadsheet & Chart as Example*

- *Practical*
  - Refactor the number guess example from yesterday to use an MVC Pattern. Should have three classes.
    - Model – A Java bean , with no other code
    - Controller , all the program logic , but no other code
    - View – a simple command line interface, does as little as possible.
  - (Optional) write a Junit test for this to replace the View
  - (Optional) write a toString() method for the JavaBean to print its contents. Log the JavaBean and see what happens.

- *Demo*
  - Example Solution, if required


Model
• Encapsulates application state
• Responds to state queries
• Exposes application functionality
• Notifies views of changes
State Query
Change Notification
State Change
View
• Renders the models
• Requests updates from models
• Sends user gestures to controller
• Allows controller to select view
View Selection
User Gestures
Controller
• Defines application behavior
• Maps user actions to model updates
• Selects view for response
• One for each functionality
Method Invocations
Events

Session 8– Introduction to Java Web Server

- *Slides*
  - What is a webserver
  - What is Tomcat
- *Demo*
  - Setting up Tomcat
    - Download
    - Unzip
    - Run
  - Setup Eclipse Project
  - Do Simple 'Hello World' JSP.
- *Practical - Editing Tomcat Projects in Eclipse*
  - ○ Setup Tomcat
  - ○ Setup Eclipse Project
  - ○ Create a new JSP. Write out the values 'Hello World / Goodbye World' as in the previous sample

Session 9 – The Struts Framework

- *Slides*
  - *(whiteboard)* Problems with JSP and Web Development
  - What is Struts?
  - What is Struts 2?

- *Demo*
  - ○Downloading Struts
  - ○Setup Eclipse Project to allow editing
  - ○Tour of the Blank Web application
    - Index.html _> Response Cycle
    - WEB-INF
      - web.xml
      - applicationContext.xml
      - lib
      - classes
      - src / java
    - META-INF

- *Practical 1*
  - Download and install Struts 2
    - Tomcat Setup
    - Editing in Eclipse
  - Take a tour of the
    - blank
    - sample application
  - Discuss the available features (about 10)
    - Structure
    - How Resuse
  - Simple changes
    - Change of property file
    - Display 2nd tag on screen
    - Write value in controller , display on screen (hello world)
- *Practical 2*

  Convert our Hello World / Goodbye world to use Struts.

  - Most simple form, ok to output loop as one piece of text

Session 10 – First Struts Application

- *Slide*
- *Demo*
- *Practical*
  - Based on Struts-Blank Web Application
  - Convert Number Guess application (MVC) to Struts Application

Session 11 – A more Sophisticated Application

- *Slide*
- *Demo*
- *Practical*
  - Add the Following features to our web application
    - A login page
    - A check to our existing page – if user is not logged in , redirect to login page
    - Internationalise our web site
    - Add Logging to our application
    - Validation on the login form

Session 12 – building our application

- *Client / Delegate Specific*

(Optional) Talking to Database

- *Slide*
  - Go through (Optional) Database slides
- *Demo*
  - Explain what Hypersonic SQL is.
  - Explain where to download it from.
  - Explain how to setup

- *Practical*
  - Practical 1: Java – SQL Code to
    - Create a Database Table
    - Insert sample data
    - Read this data back and display it
  - Practical 2 : Conversion to Web application
    - Use the code from Practical 1
    - Display the information as a web page.

Session (Optional)- ANT builds

- *Slide*
  - Whiteboard – problems with compile
      - problems with build
- *Demo*
  - Setup a simple build file
- *Practical*

Session (Optional)- Ajax and Struts

- *Slide*
- *Demo*
  - Explain what Ajax is
- *Practical*
  - Extend existing Ajax sample application.

Day 2 Conclusion

- Topics Covered in Day 2
- Recap of course aims and
- Housekeeping (evaluations)
- Feedback forms