

J2EE Web Application Development Using RAD v7.0

Course Summary

Description

This workshop will teach students to build database enabled J2EE Web programming knowledge and skills in the RAD environment using Servlets, Filters, JSP and related technology. The students will learn how to build a web application using the Model-View-Controller (or MVC or Model II) design paradigm. It will also include an overview of the Apache Struts framework. Servlets, filters and events will be covered extensively, including programming and configuring these components. All aspects of JSP will be covered. Students will write simple JSPs, write and use JavaBeans. Various aspects of accessing data and managing state efficiently are covered, including JDBC and HTTP session management.

Topics

- Using Rad
- Web Perspective
- Navigating the Java Editor
- Debug Perspective
- Servlets
- Java Server Pages
- JavaBeans
- JDBC
- Servlets and JSP Interaction
- Configuring Deployment Descriptor
- Architecting Web Application
- JSP and Servlet Filter
- Overview of Struts
- Overview of J2EE Architecture
- J2EE Web Tier Design Patterns (Optional)

Audience

This course is intended for Java developers who want to understand, design and build Web applications using the latest Java technologies.

Prerequisites

The student should have a working knowledge of Java programming and some experience with HTML.

Duration

Five days

J2EE Web Application Development Using RAD v7.0

Course Outline

I. IBM Rational Application Developer Overview

- A. Rational Application Development Platform
- B. Supported standards
- C. IBM Java IDE Release History
- D. Supported Technologies
- E. RAD Development Tools
- F. Workbench fundamentals
 - 1. Perspective
 - 2. Views
 - 3. Editors
 - 4. Project

II. Rad Web Development

- A. Web Development Tools
- B. Launching Rad
- C. Creating a Dynamic Web Project
- D. Web Perspective
- E. Web Perspective Views
 - 1. Colors View
 - 2. Gallery View
 - 3. Outline View
 - 4. Page Data View
 - 5. Palette View
 - 6. Problem View
 - 7. Properties View
 - 8. Quick Edit View
 - 9. Snippet View
 - 10. Styles View
 - 11. Tasks View
 - 12. Attributes View
 - 13. Thumbnail View
 - 14. Links View
 - 15. Servers View

III. Servlets

- A. What is a Servlet?
- B. Typical Uses of Servlets
- C. Advantages Over CGI
- D. How Servlets Work
- E. Java Servlet Architecture
- F. Servlet's Lifecycle
- G. The service() Method
- H. Writing MyServlet
 - 1. Writing HelloServlet in Rad
 - 2. Running MyServlet in RAD
 - 3. Deployment descriptor
- I. The ServletRequest Object
- J. Handling Form Data
- K. doGet() vs doPost()
 - 1. Using doGet()
 - 2. Using doPost()
- L. The ServletResponse Object

- M. Session management

IV. Using PageDesigner for HTML

- A. Page Designer
- B. Launching Page Designer
- C. Three Views of Page Designer
 - 1. Design
 - 2. Source
 - 3. Preview
- D. Creating HTML File in Rad
- E. Content Creation
- F. HTML Form Creation
- G. Updating the Action Attribute
- H. HTML Syntax Validator

V. JavaServer Pages

- A. JavaServer Pages
- B. Why use JSP?
- C. How JSP Works
- D. JSPs ILife
- E. Creating JSP in Rad
- F. Web Project Directory Structure
- G. Running JSPs in Rad
- H. JSP Tags
 - 1. Directives
 - 2. Action
 - 3. Scripting elements
 - 4. Comment
- I. The page directive
 - 1. Adding page directives in Rad
- J. Scripting elements
 - 1. Scriptlets
 - 2. Expression
 - 3. Declaration
 - 4. Comments
 - 5. Using JSPs Scripting Elements
 - 6. Adding Scripting Elements in Rad
- K. Adding include directives in Rad
- L. Implicit Objects in JSP
 - 1. Inserting Implicit Objects in Rad

VI. JavaBeans

- A. JavaBeans
- B. JavaBean structure
- C. Properties
- D. A Simple Example
- E. Standard action
 - 1. <jsp:useBean>
 - 2. <jsp:getProperty>

J2EE Web Application Development Using RAD v7.0

Course Outline (cont'd)

- 3. <jsp:setProperty>
- F. JavaBeans in JSPs
- G. Inserting JavaBeans in JSP in Rad
- H. Inserting JSP get property in Rad
- I. Inserting JSP set property in Rad
- J. Importing files in Rad
- K. Exporting files in Rad
- L. Generate setter and getter methods

VII. Integrating Servlets and JSPs

- A. Calling a servlet from JSP
 - 1. Using the FORM tag
 - 2. <jsp:include>
 - 3. <jsp:forward>
 - 4. Adding <jsp:include> in Rad
 - 5. Adding <jsp:forward> in Rad
- B. Call a JSP from a Servlet
 - 1. Using request dispatcher
 - 2. Using the sendRedirect() method of the response object
- C. SendRedirect() vs forward()

VIII. Debug Servlet and JSPs Using Rad

- A. Debugging Java Application
- B. Debug Perspective View
 - 1. Debug View
 - 2. Variables View
 - 3. Display View
 - 4. Breakpoints View
 - 5. Java Editor
 - 6. Expression View
 - 7. Console View
- C. How to Debug a Java Application
- D. Debugging a JSP
- E. Breakpoints
- F. Hit Count
- G. Launching the Debugger
- H. View Mananagement
- I. Evaluating Expressions
 - 1. Watch
 - 2. Display
 - 3. Run to Line
- J. Catching Exceptions

IX. JDBC

- A. What is JDBC
- B. Software Requirements
- C. JDBC Driver Types
- D. Basic JDBC interaction
 - 1. Load the driver
 - 2. Establish a connection
- E. Execute SQL statements
- F. Statement
 - 1. execute()

- 2. executeQuery()
- 3. executeUpdate()
- 4. ResultSet
- 5. Inserting a record
- 6. Updating a record
- 7. Deleting a record
- G. Prepared statement
- H. Callable statement
- I. DataSource
 - 1. Using a DataSource
 - 2. Define DataSource Reference
 - 3. Using the DataSource

X. Architecting Web Applications

- A. Model 1 architecture
- B. Model 2 architecture
 - 1. Model
 - 2. View
 - 3. Controller
 - 4. Advantages of model 2 architecture

XI. Deployment Descriptor – web.xml

- A. Deployment Descriptor
- B. Deployment Descriptor Editor
- C. web.xml
- D. Deployment Descriptor Diagram
- E. Wep-app Element Structure
- F. <servlet> Element Structure
- G. Adding a Servlet in web.xml
 - 1. <servlet> element for servlet
 - 2. <servlet> element for JSP
 - 3. <servlet-mapping> element
 - 4. Custom URL for a servlet
 - 5. Custom URL for a JSP
 - 6. Customizing the URL in Rad
- H. Initializing Parameters
 - 1. Adding Initializing Parameters
 - 2. Servlet Code
- I. Context parameters
 - 1. Adding Context Parameters
 - 2. Servlet Code
- J. Define Session Time Out
- K. Welcome pages
- L. Loading servlet on startup
- M. Defining error pages

J2EE Web Application Development Using RAD v7.0

Course Outline (cont'd)

XII. Overview of Struts

- A. Struts overview
- B. Overview of struts
- C. Struts framework
- D. Struts components
- E. Rad support to struts
- F. Creating a Struts Web Project
- G. Struts HTML input form
 - 1. HTML "form" tags
 - 2. The <html:form /> tagThe <html:text /> tag
 - 3. The <html:submit /> tag
 - 4. The <html:reset /> tag
 - 5. JSP Struts – an example
 - 6. Adding a JSP struts in Rad
 - 7. Populating JSP struts in Rad
- H. ActionServlet – the controller
 - 1. ActionServlet
 - 2. ActionServlet class
 - 3. How it works
 - 4. Configuring the ActionServlet
- I. The struts-config.xml
 - 1. <form-beans> element
 - 2. <action-mappings> element
 - 3. Other major elements
- J. ActionForm – the view
 - 1. ActionForm class
 - 2. ActionForm methods
 - 3. Writing AddEmployeeForm class
 - 4. Creating ActionForm in Rad
 - 5. Putting it all together
 - 6. Entry in struts-config.xml
- K. Action – the controller
- L. Action class
 - 1. The execute () method
 - 2. Writing AddEmployeeAction class
 - 3. Creating action in Rad
 - 4. Action class – an example
 - 5. Entry in struts-config.xml
- M. Validating data in the ActionForm

XIII. JSP and Servlet Filter

- A. Servlet and JSP filters
- B. What is a Filter?
- C. Typical Uses of Filter
- D. How Filters Work
- E. Filter's Lifecycle
- F. Writing the SimpleFilter
- G. Creating a SimpleFilter in Rad
- H. Deployment Descriptor
- I. Filter and Request Dispatcher
- J. Filter Mapping

XIV. Overview of J2EE Architecture

- A. J2EE architecture
- B. J2EE modules
- C. HTTP servlet
- D. JavaServer pages
- E. Model view controller
- F. Struts
- G. JSTL
- H. EJB
- I. J2EE services
 - 1. Naming service
 - 2. Database access service
 - 3. Transaction service
 - 4. Messaging service
 - 5. JavaMail service
 - 6. Security service

Optional A. J2EE Web Tier Design Patterns

- A. Design Patterns
- B. Patterns Covered
 - 1. Intercepting Filters
 - 2. Front Controller Pattern
 - 3. Context Object Pattern
 - 4. Application Controller Pattern
 - 5. View Helper Pattern
 - 6. Composite View Pattern
 - 7. Service to Worker Pattern
 - 8. Dispatcher View Pattern