

J2EE Web Development using Servlets/JSP, JDBC, and more (TT5100)

Duration: 5 days

Course Description: The J2EE platform offers all the advantages of developing in Java plus a comprehensive suite of server-side technologies. This course tells you what you need to know to design and build your own web applications. You will learn the details of the key J2EE technologies and how to leverage the strengths of each, with special focus on Servlets and JSPs. At the same time, you will be learning about the big picture of J2EE and how to design web applications that are robust, efficient, secure, and maintainable.

Audience: This is an introductory level training course, designed for experienced Java developers that need to extend their skills in web development and understand the architecture of a J2EE web application and/or implement a web application, taking advantage of what J2EE brings to the table.

Prerequisites: Attendees should have practical experience developing basic Java applications.

Topics

J2EE APPLICATION ARCHITECTURE

Technical Overview of J2EE

- What is J2EE?
- Common Themes In J2EE Framework
- J2EE Containers and Components
- Servlets
- Java Server Pages (JSP)
- EJBs and Web Services
- J2EE Containers
- J2EE Application Modules
- The J2EE 1.4 Specification
- J2EE Platform Roles

J2EE Application Architectures (web based)

- J2EE and the Web
- J2EE 1st Generation DB-centric Web Application
- Adding JSPs to Separate Presentation
- Eliminating Java code from view
- The Model 2 Architecture
- Using EJBs
- J2EE Infrastructure Supporting Web Applications Lesson Review

WEB APPLICATIONS

Understanding Web Applications

- J2EE Application Modules
- The Truth about Archives
- Enterprise Application Archive (EAR)
- Enterprise JavaBean Archive (JAR)
- J2EE Application Client (JAR)
- Resource Adapter Archive (RAR)
- Web Application Archive (WAR)
- Directory Structure

Configuring Web Applications

- Mapping an HTTP Request to a Resource
- The web.xml File
- Structure
- Declaring Servlets and JSPs
- Servlet Mapping
- Servlet Init Parameters
- Web Application init Parameters
- Welcome Page
- Error Page

J2EE Web Development using Servlets/JSP, JDBC, and more (TT5100)

DEVELOPING SERVLETS

Introduction to Servlets

- Servlet Overview
- Life Cycle of Servlets
- Servlet Lifecycle is Handled by Web Container
- HttpServlet
- Writing the init Method
- HttpServlet doXXX Methods
- Writing a Simple HTTP Servlet
- HttpServletRequest Methods
- ServletResponse
- HttpServletResponse
- Servlet I/O Classes
- Return a Status Code
- Building the Output Document
- Sending Binary Content

Processing Input Data

- Form Processing with Servlets
- HTML Form
- LoginServlet doPost

Server-Side Control

- Request Dispatcher
- Forward the processing
- Passing Processing on and Getting it Back
- Servlet Runs Within Web Container Environment
- Several Options for Receiving Data
- Init Parameters and Attributes
- ServletConfig; ServletContext
- Servlet Variables are Scoped
- HTTP Request Information
- Several Options for Sharing Data
- configuration and Context
- Servlet Variables
- HttpServlet Request
- Threading and Data

- Threading and Data 2

Client Side Control

- Output Buffering
- Setting Status Codes; Setting Headers
- sendRedirect
- Disabling Client Caching
- Supporting Persistent Connections
- Setting Content Length
- Dynamic Content Pushing

Maintaining Client State Sessions

- Management
- Tracking Problem – Stateless HTTP
- Data Problem – Data
- Solving the Tracking Problem
- Cookies
- Cookie Behavior; Retrieving Cookies
- servlet to set Cookies
- Servlet to Show Cookies
- URL Rewriting
- Solving the Data Problem
- Web Container Manages Instances
- Sessions with Cookies
- Cookie-Based Sessions
- Basic Implementation
- Cookie Detection is not Standardized
- Getting Rid of Http Sessions
- with URL Rewriting

Application and Events

- Event Listener Model
- Life Cycle Events in a Web-Application
- Declare the Listener
- Type of Events
- Context Listeners; Listeners
- Listeners for Session-Objects

J2EE Web Development using Servlets/JSP, JDBC, and more (TT5100)

FILTERS

Overview of Filters

- What is a Filter
- Single Filter
- Filter Objects
- doFilter Method
- init Method
- Filter Life Cycle
- Cascading Filters

Filtering Requests and Responses

- Request Wrapper
- Process the Request
- Examples of Request Filters
- Filter the Response
- Response Wrapper

DEVELOPING JAVASERVER PAGES

Introduction to JavaServer Pages

- Separating Presentation from Model
- Java Server Page (JSP): An Extension of Servlet
- Lifecycle of a JSP
- Example JSP
- JSP Syntax Consists of Three Types
- JSP Scripting: Declarations; Expressions; Scriptlets & Directives
- The Attribute
- The errorPage/isErrorPage Attribute
- JSP Actions
- JSP Actions: Include/Forward
- Typical JSP Access Model
- JSP Action: useBean
- Implicit Objects
- JSPs or Servlets?

JSP Implicit Objects

- Implicit Objects
- Page Object; Config Object
- Request Object; Response Object

- Out Object
- Output Buffer
- Object
- Application Object
- PageContext Object
- Attributes
- Attributes
- Exception Handling

Actions, Java Beans, and Custom Tags

- Standard Actions
- Forwarding; Including
- Using JavaBeans™ and JSP
- Declaring to use a Bean
- Using a Bean, Example
- Setting and Getting Properties
- What are Custom Tags?
- Create and Use a Custom TagLib

JSP 2.0

Introduction to JSP 2.0

- JSP 2.0 Specification
- The web.xml in J2EE 1.4
- <jsp-config>
- JSP Format Rules
- JSP Error Pages

The Expression Language

- The Expression Language
- The Expression Language (JSP 2.0)
- Enable/Disable EL
- Variables
- Literals and Operators in the JSP EL
- Implicit Objects in JSP EL
- Reserved Words
- EL Functions
- Developing the Function
- Declaring the Function in the TLD
- Using the Function
- Pre-Built EL Functions

J2EE Web Development using Servlets/JSP, JDBC, and more (TT5100)

JSTL 1.1

JSTL Introduction and Core Library

- Introduction to JSTL
- Expression Language (EL) in JSTL
- Review of JSP Bean Tags
- Tag Collaboration
- JSTL Core
- `<c:choose>` `<c:when>` `<c:otherwise>`
- `<c:if>`; `<c:import>`; `<c:forEach>`;
`<c:forEachTokens>`; `<c:out>`; `<c:param>`;
`c:catch>`; `<c:redirect>`; `<c:remove>`;
`<c:set>`;
- Using `c:set` to pass HTML
- `<c:url>`

JSTL Format Library

- JSTL Format
- `<fmt:requestEncoding>`;
`<fmt:setLocale>`; `<fmt:timeZone>`
- `<fmt:setTimeZone>`; `<fmt:bundle>`
`<fmt:setbundle>`; `<fmt:message>`
- `<fmt:formatNumber>`;
`<fmt:parseNumber>`
- `<fmt:formatDate>`; `<fmt:parseDate>`

JSTL SQL Library

- JSTL SQL
- `<sql:query>`; `<sql:update>`;
`<sql:transaction>`; `<sql:param>`
`<sql:dateParam>`

JSTL XML

- JSTL XML
- `<x:choose>` `<x:when>` `<x:otherwise>`
- `<x:out>`; `<x:if>`; `<x:forEach>`;
`<x:parse>`; `<x:set>`; `<x:transform>`;
`<x:param>`

Simple Tag Handlers

- Tag Files
- Simple Tag Handlers

- Tag File Location
- Tag-Files Outside of a Library
- Tag Library Descriptors
- The TLD File
- The `<tag-file>` Element
- Tag File with Attributes
- `jsp:attribute` Element
- Using Attributes
- `jsp:doBody`
- `jsp:invoke`

DATABASE INTEGRATION: JDBC AND J2EE

JDBC and Its Position in J2EE

- JDBC Versions
- The JDBC API
- JDBC in J2EE
- Programming with DataSources
- JNDI names
- DataSource Programming Best Practices

JDBC Data Access API

- Structured Query Language (SQL)
- Statements & Statement
- PreparedStatements
- PreparedStatement
- ResultSet
- Executing Inserts, Updates, and Deletes
- Mapping SQL Types to Java Types
- CallableStatement

The DAO Pattern

- Data Access Object (DAO)
- DAO Structure
- DAO Example: ProductDAO
- The DAOFactory
- Complete UML Diagram
- Using the ProductDAO

J2EE Web Development using Servlets/JSP, JDBC, and more (TT5100)

ADDITIONAL J2EE COMPONENTS

Working With Enterprise JavaBeans™

- Defining Enterprise JavaBeans
- JavaBeans™ vs EJBs
- EJB Architecture Overview
- EJB Container & Types of EJBs
- Enterprise Bean; Beans
- Entity Beans
- Message-Driven Bean
- EJBObject/EJBLocalObject
- Home Object (EJB Factory)
- Deployment Descriptor
- EJB-Jar File

Web Services in J2EE – WSEE

- What are Web Services?
- Web Services Architecturally
- XML and Web Service APIs
- Web Services for J2EE - WSEE
- Servlets as Web Services
- EJBs as Web Services
- Routing SOAP requests to an EJB
- WSDD

SECURITY

J2EE Security

- Typical J2EE App Server Security Services
- Java 2 Platform Security Model
- Java 2 Security in J2EE
- JAAS Authentication: Who are you?
- J2EE Security Overview
- Authorization: Are you allowed access?
- High-Level Trace of J2EE Authorization
- Deployment Descriptors Play a Large Role
- Declaring Secure Resources
- Declaring the security roles

- Typical J2EE App Server Security Services
- Security on the Web; Secure Web Traffic
- SSL In Action
- Responsibilities For Security
- CMS: Declaring HTTPS
- Authentication Challenge Mechanisms

Web Application Security Overview

- Attacks are Constant and Changing
- Open Web Application Security Project
- Assets are the Targets
- The Context for Defensive Coding
- Attackers Not Hackers
- Cross-Site Scripting (XSS): Description
- Defending Against XSS Attacks
- SQL Injection: Description; Example & Drill Down
- Defending Against SQL Injection
- Attacks

Handling Untrusted Input

- Unvalidated Input: Description
- Protecting a Web Resource
- Defending a Web Application
- Defending a Web Application/Resource
- Responding to Error State
- Best Practices for Untrusted Data
- Additional Types of Attacks

ADDITIONAL J2EE TOPICS

Transactions

- Transaction Definitions
- The ACID Transaction Properties
- Transaction Lifecycle
- Overview of a Transactional System

J2EE Web Development using Servlets/JSP, JDBC, and more (TT5100)

- J2EE Transaction Support

Other J2EE APIs

- Java Message Service (JMS)
- When is Messaging Used?
- Two Messaging Models
- More On Publish/Subscribe
- Logical View of Publish/Subscribe
- More On Point-to-Point (P2P)
- Logical View of Point-To-Point
- Message Servers
- JavaMail; JavaMail Architecture
- XML
- An XML Document
- J2EE and XML

The J2EE Blueprints

- Overview
- Key Resources

APPENDIX: CUSTOM TAGS

Introduction to Custom Tags

- Components that Make up a Custom Tag
- Example of a TagHandler
- Tag Library Descriptor (TLD)
- The Tag Element
- Attribute Element; Variable Element
- Example Tag Library JAR
- Example WAR
- Example Using the Tag in a JSP
- Viewing the Result

DEVELOPING CUSTOM TAGS

- Package overview
- Developing the TagHandler
- TagSupport Sequence
- Iterations Without Generating bodycontent
- The doStartTag
- Programming the doEndTag