

Course Description: This course introduces the students to Jakarta Struts 1.2. During the course, students will learn to design and develop Struts based applications. Students learn how to incorporate JSPs, servlets, EJBs and JavaBeans into their design. Corresponding to every chapter, there is a lab reinforcing the concept.

Who Should Attend: This course is intended for programmers and designers who want to design and develop applications using the Jakarta Struts Model-View-Controller (MVC) framework.

Prerequisites: Students should have a good understanding of HTML and object-oriented programming using Java. Students should also have experience with JSPs and servlets with some understanding of JavaBeans or EJBs.

Benefits of Attendance: Upon completion of this course, students will be able to:

- Understand and explain the Jakarta Struts framework
- Design and build Struts based applications
- Use MyEclipse to create a simple Struts application
- Develop a simple custom tag lib
- Use The Struts Tag Lib
- Use the DynaActionForm and Validator feature of Struts 1.2.

Course Outline:

Struts Architecture and Overview

Review: MVC Model
Review: Request, Session and Application
What is Struts?
Struts Flow
Components in Struts
The Model
The View
The Controller
Struts Flow
struts-config.xml
struts-config.xml Content
The <data-sources> element in the struts-config.xml
The <form-beans> element in the struts-config.xml
The <global-forwards> element in the struts-config.xml
The <action-mappings> element in the struts-config.xml
Sample struts-config.xml
The Role of web.xml for the Application
Configuring Struts in the web.xml
web.xml Content
Steps to Configuring Struts in the web.xml

The First Struts Application

The application
The Structure
Setup Needed
Directory Structure
index.jsp
The index.jsp Page
struts-config.xml
MultiplyForm.java
MultiplyAction.java
result.jsp
The result page
ApplicationResources.properties

Development of Struts Applications

Using RAD V7
Directory Structure of a Struts Application
RAD Struts Support
Create a Struts Web Application Project
View the Struts Project Structure
Create a Struts JSP
Insert Struts Tags
Creating a Struts Form Bean
Create a Struts Action
Edit a Struts Configuration File

Struts Development Cycle

Gathering Requirements
Defining Screen Requirements
Detailed screen design
Determining the Screen Flow
Defining the ActionMappings in the struts-config.xml

Struts view components – Data vs. ActionForm Bean
Defining Screen Requirements – define FormBean
Developing the FormBean
Developing the FormBean – The reset() Method
Developing the FormBean – The validate() Method
Developing the FormBean – ActionErrors
Developing the FormBean – Defining the Message Key
Developing the Action Class – action and Action Class
Developing the Action Class
Developing Actions – The execute() Method
Developing Actions – ActionForward
Developing Actions – ActionMapping class
Developing Actions – execute() example
execute() Method Example:
Developing Business Logic – EJBs
Developing JSPs
Configuring struts-config.xml and web.xml
Build, Pack, and Deploy

Struts Tag Libraries

Commonality among the Struts Tags
Bean Tags
HTML Tags
Logic Tags
Logic Tags functionality

Struts Extensions, Internationalization and Error Handling

Struts Extensions
Extension Points
Plug-ins
Custom Configuration Class
Writing a Configuration Class
Custom ActionServlet
Custom RequestProcessor
Base Action Class
Base Form Bean
Custom JSP Tags
Internationalization (I18N)
Error Handling
The ActionError class
Error Handling : validate() method of ActionForm
Displaying the errors found in the validate method
Error Handling : execute() method of Action
Declarative Exception Handling
Syntax of declarative exception handling
Programmatic Exception Handling
Logging from Struts

Using Commons Logging and Log4J in Struts
Writing Commons Logging Code

Miscellaneous Advanced Features

Integration of Jakarta Common Libraries - BeanUtils
Integration of Jakarta Common Libraries - Digester
Multiple Sub-application Support
DynaActionForms
Validators
Adding the Validator framework to a Struts application
A tale of three files
A sample rule in the file validator-rules.xml
validation.xml file
Validator Dependency
Rule Variables
Error Message
Basic Validation Rules
Client Side Validation
Validating multiplication example using DynaActionForm
Writing Custom Validators
Validator Class Example
HTTP Redirection
Working With Check Boxes
Context Sensitive Form Validation

Database Programming

Basic Concepts
MVC Interaction
Database Connection
Transaction Management
Data Source
Defining a Struts Data Source
Opening a Connection from a Struts Data Source
Creating an Editor Form
Example Form Display Action

Templates and Tiles

Struts Templates
Defining the Template
A template
Using the template in a JSP
<template:insert>
<template:put>
A JSP that uses the template
Struts Templates
Templates as UI components
Struts 1.2 Tiles Support
Tiles
A JSP that uses the layout
Tiles go beyond templates
Inheritance in tiles definitions
Definitions as Struts "forwards"
Template or Tiles?

Unit Testing Struts Applications

What is JUnit?
Why JUnit?
A JUnit Test
Running the tests
JUnit Basics
Unit Testing Struts Applications
A simple Struts test case
MockStrutsTestCase Methods
Downloading STC
Testing Strategies

JSP Expression Language and Standard

Tag Library
JSP Expression Language (EL)
Basic Usage
Built-in Objects
Working With Arrays and Maps
Operators
Full Example
JSP Standard Tag Library (JSTL)
Run Time Version
Basic Tags
Condition Tags
Iterator Tags
Internationalization (I18N)
Setting Preferred Locale
Specifying Resource Bundle
Display Translated Text
Display Number
Display Date
JDBC Tags
Specify Data Source
Performing a Query
Display Result
Pagination Example