Updated on 19/06/2023

Sign up

JSF Training

3 days (21 hours)

Presentation

JavaServer Faces is a Java framework for developing Web applications.

Unlike other traditional action-based MVC frameworks, JSF is based on the notion of components, comparable to Swing or SWT, where the state of a component is saved when the page is rendered, and then restored when the request is returned.

JSF is presentation technology agnostic. It uses Facelets by default since version 2.2, but can be used with other technologies, such as JSP or XUL.

JavaServer Faces is an MVC web framework that simplifies the construction of user interfaces for server-based applications using reusable UI components within a page. JSF offers the ability to connect UI widgets to server-side data sources and event handlers.

Objectives

- Learn to use JSF - Discover the essential possibilities of this Framework
- Be able to develop a web application integrating a user interface based on standard JSF technology integrated into Java EE
- Understanding JSF concepts
- Implementing JSF graphical components in a web page
- Connect JSF graphical elements to data
- Allows you to confront the classic problems of web applications and to assess the solutions provided by JSF.

Target audience

- Architects,Project managers,Engineers,Programmers,Developers

Prerequisites

- Knowledge of servlets and JSP
- Practical Java EE development and)
- Basic knowledge of the Internet and HTML

Program

Environment configuration

- System requirements
- Configuring the environment for JSF application development
  - Check that Java is installed on your machine
  - Configuring the Java development kit
  - Configuring Eclipse IDE
  - Download Maven archive
  - Extracting the Maven archive
  - Setting Maven environment variables
  - Add Maven bin directory location to system path
  - Checking Maven installation
  - Configuring Apache Tomcat

Architecture

- What is MVC Design Pattern?
- JSF architecture
  - JavaBeans components as templates containing application-specific functions and data
  - A library of custom tags to represent managers and
    event validators
  - A library of custom tags for rendering user interface components
  - User interface components represented as status objects on the server
  - Server-side help classes
  - Validators, event handlers and navigation managers
  - Application configuration resource file for configuring application resources

Lifecycle

- Phase 1: Restoring the view
- Phase 2: Apply demand values
- Phase 3: Process validation
- Phase 4: Updating model values
- Phase 5: Request for intervention
- Phase 6: Delivering the answer

First request

- Create a project
- Add JSF capability to the project
- Preparing the Eclipse project
- Importing a project into Eclipse
- Configuring the Faces servlet in web.xml
- Create a managed bean
- Creating a JSF page
- Building the project
- Deploy WAR file
- Run the application

Beans under management

- Using XML configuration
- Using annotation
- Staff annotations

Page navigation

- Implicit navigation
- Automatic JSF page navigation
- Automatic navigation in Managed Bean
- Conditional navigation
- Action-based navigation resolution
- Forward vs Redirect
- Basic tags
- Faceted tags
- Converter tags
- Validator tags

DataTable

- HTML header

Composite components

- Define custom component
  - Create a custom component
  - Using composite labels
- Using the custom component
  - Using a custom namespace
  - Using a personalized label

Ajax and JSF

- Introduction to Ajax, Matching the two models
- Ajax support in JSF 2.0
- Introducing the RichFaces and ICEFaces frameworks
- The emergence of mobile libraries

Event management

- Principles and types of FacesEvent events
- ActionEvent management
- ValueChangeEvent management
- DataModelEvent management
- New events in JSF 2.0

JDBC integration

- SQL database commands

Expression Language

- Operating principle, evaluation engine
- Elements of syntax
- Implicit objects
- The different contexts

Spring integration

- Add a DelegatingVariableResolver
- Add contextual listeners
- Define dependence
- Add a dependency
- Dependency on use

Internationalization

- Define property files
- Update faces-config.xml
- Using bundle var resources

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.