

JavaServer Faces

This article will take you through the basics of Java Server Faces. Later we will have samples which will elaborate the features and functionalities of Java Server Faces.

Introduction: Java Server Faces (hereafter JSF) is an event based development framework in Java EE5 arena. It is a framework for building user interfaces for web application. JSF follows a MVC architecture, web application following MVC architecture are best developed using JSF.

Let us briefly understand what is a MVC architecture. Any Enterprise Application has a Business component, Presentation component and a Controller component. Usually small applications will have all the components tightly coupled with each other. Any change to one component will have an effect on the other. As the size of the application increases it becomes more difficult and complex to debug, modify the application. So as to solve this problem we need to improve the architectural design of the application.

Model(M), View(V) and Controller(C) is one step forward towards improving the application design.

- Model handles data and logic of the application. It handles the business component of the application.
- View takes care of the User interface. It handles the output of the application that has to be displayed to the user. It handles the presentation component.
- Controller handles the input from user and modifies the model or view according to the user input. A controller selects the next view that has to be displayed on the basis of the user input and outcome of the model operations.

By following a MVC architecture there is a division of functionalities among Model, View and Architecture. This model has got the following advantages

- Helps to reduce duplication in the code
- Maintenance of the existing code becomes easier
- Provides a central place to control all the user views- Controller
- Any new user view can be added and later wired to the controller
- Debugging the application becomes quite easier
- Changes to the Model have a minimal effect on View and Controller

Coming back to JSF now, JSF has the following as the major component

1. User Interface Components- These are stateful objects which are built over JavaBeans. Each UI component has associated events, properties and methods. These component reside on server side rather than having them on the client side(for example swing components). In JSF UI components are rendered for the browser. For any user request components output the HTML, which is then rendered and shown to the user as GUI elements.
2. Navigation- JSF defines a very effective navigation system. For a given page the navigation rules defines what are the valid outcomes and which page to load for each output.
3. Renderer- This component is used to render UI components. This component can provide different rendering options depending on different types of browser (example Netscape, Internet explorer, Mozilla, Firefox). A renderer translates the user input to values which are understandable by components and vice versa.
4. Converter- A converter converts data from one type to another. The converted in JSF converts user inputs to components and vice versa.
5. Validator- A validator validates the user input and verifies whether the input is in acceptable format. In the conventional web development validation is taken care by scripting languages like Java Script and that puts a considerable overhead over the application. JSF provides the controller mechanism which validates the input. Validation is done at the UI component level as well as the bean level.
6. JavaBeans- In JSF Java Beans define the interaction between UI component and model. They play the role of Controllers in JSF. We can say that Backing Bean controls the form and the components associated with the form.
7. Internationalization and Localization- JSF uses the internationalization tags from Java Server pages standard Tag library(JSTL).

This completes what gives the basic idea on architectural design of Java Server Faces and the various components available with Java Server Faces. Underlining samples will describe how MVC architecture and various components are being used in application development.

- [Developing a Simple JavaServer Faces application](#)