JSP JSF SERVLET DESCRIPTIVE

1. JSF Features

Latest version of JSF 2.2 provides the following features.

Component Based Framework

Implements Facelets Technology

Integration with Expression Language

Support HTML5

Ease and Rapid web Development.

Support Internationalization

Bean Annotations

Default Exception Handling

Templating

Inbuilt AJAX Support

Security

2. Benefits of JavaServer Faces

1) It provides clean and clear separation between behavior and presentation of web application. You can write business logic and user interface separately.

2) JavaServer Faces API?s are layered directly on top of the Servlet API. Which enables several various application use cases, such as using different presentation technologies, creating your own custom components directly from the component classes.

3) Including of Facelets technology in JavaServer Faces 2.0, provides massive advantages to it. Facelets is now the preferred presentation technology for building JavaServer Faces based web applications.

3. JavaServer Faces Lifecycle

The JSF lifecycle is divided into two main phases:

Execute Phase

Render Phase

1) Execute Phase

The execute phase is further divided into following subphases.

Restore View Phase

Apply Request Values Phase

Process Validations Phase

Update Model Values Phase

Invoke Application Phase

Render Response Phase

2) Render

In this phase, the requested view is rendered as a response to the client browser. View rendering is a process in which output is generated as HTML or XHTML. So, user can see it at the browser.

The following steps are taken during the render process.

Application is compiled, when a client makes an initial request for the index.xhtml web page.

Application executes after compilation and a new component tree is constructed for the application and placed in a FacesContext.

The component tree is populated with the component and the managed bean property associated with it, represented by the EL expression.

Based on the component tree. A new view is built.

The view is rendered to the requesting client as a response.

The component tree is destroyed automatically.

On subsequent requests, the component tree is rebuilt, and the saved state is applied.