

JSF Interview Questions

A list of top frequently asked JSF interview questions and answers are given below.

1) What is JSF (JavaServer Faces)?

It is a server-side component-based user interface framework. It is used to develop web applications. It provides a well-defined programming model and consists of rich API and tag libraries. The latest version JSF 2 uses Facelets as its default templating system.

For more information: [Click here.](#)

2) What are the benefits of JSF (JavaServer Faces)?

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

For more information: [Click here.](#)

3) What are the features of JSF (JavaServer Faces)?

The 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

For more information: [Click here.](#)

4) Give the system requirements for the JSF use.

For using JSF in a computer, the platform needs to have JDK 1.5 or above.

5) What is the life cycle of JSF (JavaServer Faces)?

JavaServer Faces application framework manages lifecycle phases automatically for simple applications and also allows you to manage that manually. The lifecycle of a JavaServer Faces application begins when the client makes an HTTP request for a page and ends when the server responds with the page.

For more information: [Click here.](#)

6) How does the execution phase work in JSF (JavaServer Faces) life cycle?

In the execute phase, when the first request is made, the application view is built or restored. For other subsequent requests other actions are performed like request parameter values are applied, conversions and validations are performed for component values, managed beans are updated with component values, and application logic is invoked.

For more information: [Click here.](#)

7) How does render phase work in JSF (JavaServer Faces) life cycle?

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, the user can see it on the browser.

For more information: [Click here.](#)

8) What is managed bean in JSF (JavaServer Faces)?

It is a pure Java class which contains a set of properties and the set of the getter, setter methods.

Following are the common functions that managed bean methods perform:

- Validating a component's data

- Handling an event fired by a component
- Performing processing to determine the next page to which the application must navigate

For more information: [Click here](#)

9) How to configure managed bean in the XML file?

In this case, we configure bean-name, bean-class, and bean-scope in the XML file to make it accessible in the project. This is an older approach to configure bean into an XML file.

For more information: [Click here.](#)

10) What are the available scopes for the managed bean?

You can use following scopes for a bean class:

- **Application (@ApplicationScoped):** Application scope persists across interaction of all the users with the web application.
- **Session (@SessionScoped):** Session scope persists across multiple HTTP requests in a web application.
- **View (@ViewScoped):** View scope persists during the interaction of the user with a single page (view) of a web application.

For more information: [Click here.](#)

11) What is eager managed bean?

The Managed bean is lazy by default. It means the bean is instantiated only when a request is made from the application.

You can force a bean to be instantiated and placed in the application scope as soon as the application is started.

For more information: [Click here.](#)

12) What are User Interface Components in JSF (JavaServer Faces)?

JavaServer Faces HTML tag library represents HTML form components and other basic HTML elements, which are used to display or accept data from the user. A JSF form sends this data to the server after submitting the form.

For more information: [Click here.](#)

13) What are the MVC design modules?

- Model
- View
- Controller

14) Give a list of facelet tags.

- Templates
- Parameters
- Custom
- Remove

15) Give a list of important DataTable operations.

- Display DataTable
- Add data
- Edit data
- Delete data
- Using the data model

16) What does AJAX stand for?

AJAX stands for Asynchronous JavaScript and XML.

17) List some AJAX attributes.

- Disabled
- Event
- Execute
- Immediate

- Listener
- Onerror
- Onevent
- Render

18) Name some important event handlers.

- ActionListener
- ApplicationEvents
- ValueChangeListener

19) What are the requirements for using integrated JSF using JDBC?

- PostgreSQLJDBC4 Driver
- PostgreSQL 9.1

20) What is the use of Controller module?

It handles the processing of an application.

21) What is the use of the View module?

The module shows the user interface.

22) What is h:inputText tag in JSF (JavaServer Faces)?

The JSF <h: inputText> tag is used to render an input field on the web page.

It is used within a <h: form> tag to declare input field that allows the user to input data.

For more information: [Click here.](#)

23) What is h:outputText tag in JSF (JavaServer Faces)?

The JSF `<h:outputText>` is used to render a plain text. If the "styleClass", "style", "dir" or "lang" attributes are present, render a "span" element. If the "styleClass" attribute is present, render its value as the value of the "class" attribute.

For more information: [Click here.](#)

24) What is h:form tag in JSF (JavaServer Faces)?

The `<h:form>` tag represents an input form. It includes child components that can contain data which is either presented to the user or submitted with the form. It can also include HTML markup to lay out the components on the page.

For more information: [Click here](#)

25) What is h:commandButton tag in JSF (JavaServer Faces)?

The `<h:commandButton>` tag creates a submit button and used to submit an application form.

For more information: [Click here.](#)

26) What is h:inputTextarea tag in JSF (JavaServer Faces)?

The `<h:inputTextarea>` tag renders an HTML "textarea" element. It allows a user to enter multiline string.

For more information: [Click here.](#)

27) What is h:commandLink tag in JSF (JavaServer Faces)?

JSF renders it as an HTML "a" anchor element that acts as a form submit button when clicked. So, you can create an anchor tag by using this tag. An `h:commandLink` tag must include a nested `h:outputText` tag, which represents the text that the user clicks to generate the event. It's also required to be placed inside a `<h:form>` tag.

For more information: [Click here.](#)

28) What is h:inputSecret tag in JSF (JavaServer Faces)?

It is a standard password field which accepts one line of text with no spaces and displays it as a set of asterisks as it is entered. In other words, we say, it is used to create an HTML password field which allows a user to input a string without the actual string appearing in the field.

For more information: [Click here.](#)

29) What is h:inputHidden tag in JSF (JavaServer Faces)?

It renders an HTML "input" element of type hidden. It does not appear on the web page, so you can pass hidden information while submitting the form.

For more information: [Click here.](#)

30) What is h:inputFile tag in JSF (JavaServer Faces)?

JSF renders it as an HTML element of type file. It is used to get the file as input. In HTML form, it allows a user to upload a file.

For more information: [Click here.](#)

31) What is h:graphicImage tag in JSF (JavaServer Faces)?

JSF renders an HTML element "img" tag. This tag is used to render an image on the web page.

For more information: [Click here.](#)

32) What is h:message tag in JSF (JavaServer Faces)?

It is used to display a single message for a particular component. You can display your custom message by passing the id of that component into the for attribute.

For more information: [Click here.](#)

33) What is h:messages tag in JSF (JavaServer Faces)?

It is used to displays all messages that were stored in the faces context during the course of the JSF lifecycle.

For more information: [Click here.](#)

34) What is h:dataTable tag in JSF (JavaServer Faces)?

It is used to create a data table that can be updated dynamically.

For more information: [Click here](#)

35) What are the available validation tags in JSF (JavaServer Faces)?

JavaServer Faces technology provides a set of standard classes and associated tags that you can use to validate elements data. A table which contains the validation tags is given.

For more information: [Click here.](#)

36) What is f:validateBean tag in JSF (JavaServer Faces)?

It is used to register a bean validator to the component. For validating the bean model, you must set the context parameter in the web deployment descriptor file web.xml.

For more information: [Click here.](#)

37) What is f:validateDoubleRange tag in JSF (JavaServer Faces)?

It is used to check that the value of an input field is within a specific range or not. The value must be a float or double type.

For more information: [Click here.](#)

38) What is f:validateLength tag in JSF (JavaServer Faces)?

It is used to check whether the length of a component's value is within a specific range or not. The value must be a `java.lang.String`.

For more information: [Click here.](#)

39) What is f:validateLongRange tag in JSF (JavaServer Faces)?

It is used to check whether the local value of a component is within a specific range or not. The value must be any numeric type or `String` that can be converted to a `long`.

For more information: [Click here.](#)

40) What is f:validateRegex tag in JSF (JavaServer Faces)?

It is used to check whether the local value of a component is a match against a regular expression from the `java.util.regex` package or not.

For more information: [Click here.](#)

41) What is f:validateRequired tag in JSF (JavaServer Faces)?

It is used to ensure that the local value is not empty on an `EditableValueHolder` component.

For more information: [Click here.](#)

42) How to validate managed bean in JSF (JavaServer Faces)?

JSF provides validation constraints for bean model in the form of annotations. You can place those annotations on a field, method, or class of a `JavaBeans` component, such as a managed bean.

For more information: [Click here.](#)

43) What are the standard converters in JSF (JavaServer Faces)?

The JavaServer Faces provides a set of Converters. You can use that to convert component data. The purpose of conversion is to take the String-based data from the Servlet API and convert it to strongly typed Java objects.

For more information: [Click here.](#)

44) What is f:converter tag in JSF (JavaServer Faces)?

It is a core converter tag. It is used to add an arbitrary converter to the parent component.

For more information: [Click here.](#)

45) What is f:convertDateTime tag in JSF (JavaServer Faces)?

It is used to convert user input into the specified date. You can convert a component's data to a `java.util.Date` by nesting the `convertDateTime` tag inside the component tag. The `convertDateTime` tag has several attributes that allow you to specify the format and type of the data.

For more information: [Click here.](#)

46) What is f:convertNumber tag in JSF (JavaServer Faces)?

It is used to convert component (user input) data into a Java Number type. You can convert a component's data to a `java.lang.Number` by nesting the `convertNumber` tag inside the component tag. The `convertNumber` tag has several attributes that allow you to specify the format and type of the data.

For more information: [Click here.](#)

47) How to refer bean method in JSF (JavaServer Faces)?

We refer a managed bean method that performs navigation processing for the component and returns a logical outcome String.

For more information: [Click here.](#)

48) What is Facelets?

It is a lightweight page declaration language which is used to build JavaServer Faces views using HTML style.

For more information: [Click here.](#)

49) What are the advantages of Facelets?

- 1) It supports code reusability through templating and composite components.
- 2) It provides functional extensibility of components and other server-side objects through customization

For more information: [Click here](#)

50) What is the lifecycle of Facelets application in JSF (JavaServer Faces)?

The JavaServer Faces specification defines the lifecycle of a JavaServer Faces application. The following steps describe that process to a Facelets-based application.

- 1) Lifecycle starts when a client makes a new request for a web page which is created using Facelets. JSF creates a new component tree or `javax.faces.component.UIViewRoot` and placed into the `FacesContext`.

For more information: [Click here](#)

51) How to create a Facelet view?

Facelets views are XHTML pages. You can create a web page or view, by adding components to the page, wire the components to backing bean values and properties, and register converters, validators, or listeners on the components.

For more information: [Click here](#)

52) How to map Faces Servlet instance in JSF (JavaServer Faces) application?

The configuration of a JavaServer Faces application is done by mapping the Faces Servlet in the web deployment descriptor file a web.xml.

For more information: [Click here](#)

53) What are the Facelets Templates?

It is a tool which provides the facility to implement the user interface. Templating is a useful Facelets feature that allows you to create a page that will act as the base for the other pages in an application. By using templates, you can reuse code and avoid recreating similar pages again and again.

For more information: [Click here](#)

54) How to create Facelets Templates?

Templating is a useful Facelets feature that allows you to create a page that will act as the base for the other pages in an application.

For more information: [Click here](#)

55) What are the Facelets Composite Components?

JSF provides the concept of composite components with Facelets. The Composite component is a special type of template that acts as a component in your application.

For more information: [Click here](#)

56) What are web resources in JSF (JavaServer Faces)?

JSF web resources are the resources which are required for proper rendering in the web application. It includes images, script files, and any user-created component libraries.

For more information: [Click here](#)

57) How to access CSS (Cascading Style Sheets) File in JSF (JavaServer Faces) application?

The **<h:outputStylesheet>** tag is used to access CSS (Cascading Style Sheets) resource in the web application. You must create a subdirectory inside the resources folder.

For more information: [Click here](#)

58) How to access JS (JavaScript) File in JSF (JavaServer Faces) application?

The **<h:outputScript>** tag is used to access JavaScript file in the web application. You must create a subdirectory inside the resources folder.

For more information: [Click here](#)

59) How to relocate web resources in JSF (JavaServer Faces) application?

JSF provides a facility to place your resources at any section of your web page and render it to another section. You can relocate your resource by specifying the target attribute.

For more information: [Click here](#)

60) How to create JDBC (Java Database Connectivity) connection in JSF (JavaServer Faces) application?

You can integrate JSF application to the JDBC. JDBC allows you to store data into the database table.

For more information: [Click here](#)