

APRIL 2, 2018 BY PANKAJ — 10 COMMENTS

JSF Interview Questions



1. What is JSF?
2. What is a Managed Bean?
3. What are the three types of text fields tags provided by JSF?
4. What is the significance of @ManagedProperty annotation?

5. What does @ApplicationScoped annotation indicate?
6. What is Resource bundling in JSF?
7. Explain the required and requiredMessage attribute of the <h:inputText> tag?
8. What are the different types of Page Navigation supported in JSF?
9. What are JSF life cycle phases?
10. Explain some of the attributes of <h:form> tag?
11. What are the command component tags used for action and navigation?
12. What are Data Bound table components ?
13. What is an event?
14. How can we obtain the generated event?
15. What are the different types of JSF events?
16. What is a listener class?
17. What is the significance of facelets tag?
18. What are some of the facelets tags?
19. What are the different types of validations in JSF?
20. What are different types of expressions supported by JSF EL?
21. What are immediate and deferred value expressions?
22. Explain value expression and method expressions?
23. Explain @ViewScoped, @SessionScoped, @CustomScoped and @RequestScoped annotations?
24. Explain different ways of declaring a managed bean in JSF?
25. What is the significance of name and eager attributes in managed bean?
26. Mention some of the validator tags used in JSF?
27. What are the benefits of using JSF Framework?
28. What are different JSF Converter tags?
29. List the benefits of Expression Language?
30. What is a backing bean?
31. What are standard JSF tag libraries?
32. Mention some of the functions that the backing bean method performs?
33. What are different implementations of JSF API?
34. Explain JSF architecture?
35. How different components are rendered in JSF page?
36. Can the JSF support multiple faces configuration files?
37. What are the differences between a Backing Bean and Managed Bean?
38. How to display error messages in JSF?
39. What is the significance of selectOne menu tag in JSF?
40. Explain immediate and rendered attributes?
41. Mention two ways of binding supported by JSF?
42. What are required configurations for JSF framework?
43. What is JSF Navigation Rule?
44. What is the role of JSF converter tags?
45. List the benefits of data table tags in JSF?
46. How to implement internationalization (i18n) in JSF?
47. What is component rendering model?
48. What is a render kit?

49. [What is view object?](#)
50. [What do you mean by Bean Scope?](#)
51. [What is the difference between JSF-1 and JSF-2?](#)
52. [Can we have a JSF application without faces-config.xml?](#)
53. [What are some of the best practices for JSF application?](#)
54. [How do you compare JSF with Spring Framework?](#)
55. [Why JSF is not so popular as MVC framework like Spring MVC, even though its older?](#)
56. [Can we integrate JSF with other popular frameworks such as Spring, Hibernate etc?](#)

JSF Interview Questions and Answers

1. What is JSF?

Java Server Faces (JSF) technology is a front end framework which makes the creation of user interface components easier by reusing the UI components. JSF is designed based on the Model View Controller pattern (MVC) which segregates the presentation, controller and the business logic.

For getting started with JSF, please read [JSF Tutorial for Beginners](#).

2. What is a Managed Bean?

A managed bean is a java class registered to JSF which makes interaction between the UI and the business logic possible. Managed Beans can be created using @ManagedBean annotation. For a detailed example, please read [JSF Managed Bean Example](#).

3. What are the three types of text fields tags provided by JSF?

The three types of text field tags are;

1. <h:inputText> – This adds the text box next to the label field.
2. <h:inputSecret> – This type is used for password fields where the entered data is hidden.
3. <h:inputTextarea> – This type of fields is used while entering large number of characters.

For complete example, please go through [JSF Text Components](#).

4. What is the significance of @ManagedProperty annotation?

The `@ManagedProperty` annotation enables us to inject a managed bean into another managed bean. To learn how it works, please read [JSF Injecting Managed Bean](#).

5. What does `@ApplicationScoped` annotation indicate?

The `@ApplicationScoped` annotation indicates that the bean is valid as long as the web application is valid.

6. What is Resource bundling in JSF?

The phenomenon of storing the UI labels, date, status messages and other UI textual elements in a separate properties file instead of hardcoding these in a page is called resource bundling.

We can use `h:outputLabel` element to pick these values from resource bundle properties file in JSF view pages, for a complete example please refer [JSF Resource Bundle Example](#).

7. Explain the required and requiredMessage attribute of the `<h:inputText>` tag?

Required attribute indicates that the field is mandatory when set to true. The `requiredMessage` attribute allows users to specify their own message for the ui components when the fields are mandatory. They are used for declarative validations in JSF view pages, for a complete example please read [JSF Declarative Validation](#).

8. What are the different types of Page Navigation supported in JSF?

The types of Page navigation supported in JSF are

1. Implicit Navigation
2. Navigation through Managed Bean
3. Navigation through faces-config.xml
4. Forward versus Redirect navigation
5. Conditional Navigation

For detailed explanation, please go through [JSF Navigation Rules](#) and [JSF Action Method Navigation Rule Example](#).

9. What are JSF life cycle phases?

There are six lifecycle phases namely;

1. Restore view phase
2. Apply request values phase
3. Process validations phase
4. Update model values phase
5. Invoke application phase

6. Render response phase

For better understanding and detailed explanation of each of life cycle phase, please read [JSF Page Lifecycle Management](#).

10. Explain some of the attributes of <h:form> tag?

Some of the important h:form tag attributes are;

- id: This is the unique identifier used to identify a component.
- title: A title for an element of the form used as tooltip.
- onclick: invokes the javascript function to be called when a button is clicked next to an element.
- onsubmit: invokes javascript function to be called on click of form by a submit button.
- onreset: Javascript to be invoked on the reset of the elements in a form.
- ondblclick: Javascript code to be executed when the mouse is double clicked over a field in a form.
- onmouseup: Javascript code to be executed when the mouse button is released over a component.
- onmousedown: Javascript code to be executed when the mouse pointer is clicked down over this element.
- binding: value of the expression linked to a property in a backing bean.
- target: Name of the frame where the resource retrieved is to be displayed.
- accept: the contents list that the form can handle.
- acceptCharset: defines the list of character encoding that the form will accept.
- style: The CSS style definitions that can be applied for the form
- prependId: flag that indicates whether id should be prepended to the form
- dir: Overrides default text functionality for this component.

To see form component in action, please go through [JSF Form Component Example](#).

11. What are the command component tags used for action and navigation?

The command component tags for performing action and navigation are

1. <h:commandButton> tag – The h:commandButton tag renders a button to submit a form thereby paving a way for processing the data entered by the user.
2. <h:commandLink> tag – The commandLink provides an hyperlink equivalent to anchor tag in HTML that acts like a submit button and can be associated with the backing beans or action class for event handling.

For detailed example, please go through [JSF Command Components Example](#).

12. What are Data Bound table components?

The components that are responsible for displaying the relational data in a tabular format are called data bound table components. The <h:dataTable> tag is used for displaying the data components. The

<h:column> tag iterates over each record in the data source displayed in rows.

Some of the attributes of the h:dataTable tag are;

- bgcolor: background color for the table that is displayed.
- border: width in pixel to be drawn around the table.
- cellpadding: Space between border of each cell and its contents.
- cellspacing: Space between left side of the table and leftmost column and also amount of space between the cells.
- columnClasses: List of css styles separated by comma to be applied to the columns of this table.
- bodyrows: List of row indices separated by comma to be applied for the "tbody" element should be started.

To see dataTable tag in action, please read [JSF dataTable tag](#).

13. What is an event?

An event is defined as a signal triggered based upon the user actions such as click of button, hyperlink, changing the input value etc. JSF tells the component to invoke the appropriate listener class that process the event generated by the user.

14. How can we obtain the generated event?

The generated event can be obtained by calling event.getComponent as

```
UIComponent ui = new UIComponent();  
MyFacesEvent ev1 = new MyFacesEvent(ui);  
UIComponent sc1 = ev1.getComponent();
```

15. What are the different types of JSF events?

There are three types of JSF events namely

1. Action Events: Action events are the events that are generated for the ui components like command button or command hyperlink.
2. Value Change Events: Value change events refers to the UI components textfield, radio button, list box etc. The value change event will get fired as soon as the value is changed in the UI component.
3. Phase Events: This type of event involves the events to be fired in one of the six phases of JSF lifecycle either during start or towards the end of each phase.

You can get more details about these events at [JSF Events Example](#).

16. What is a listener class?

A class which is associated with an event is called a listener class. For example, if the event is a valueChange event then the corresponding listener class **ValueChangeListener** is associated with it. Read more at [JSF Listeners Example](#) to learn more about these listeners.

17. What is the significance of facelets tag?

JSF provides a special set of tags that gives the flexibility to manage common tags/parts in one place for more than one application. These tags allow us to create a common layout that can be used across applications. You can include facelets tags using below code;

```
<html
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:ui="http://java.sun.com/jsf/facelets"
>
```

I have written a very long article for facelets tags, go check it out at [JSF Facelets Tags Example](#).

18. What are some of the facelets tags?

Some of the important facelets tags are;

- <ui:component> tag
- <ui:composition> tag
- <ui:decorate> tag
- <ui:define> tag
- <ui:fragment> tag
- <ui:include> tag

19. What are the different types of validations in JSF?

There are two types of validations namely;

1. **Declarative Validations:** The validations that are fired using JSF standard validators or Bean validators fall under declarative type.
2. **Imperative validation:** The standard validation messages would not be sufficient in all the cases and sometimes may require complex validations that are declared by the user overriding the standard validations and these are called Imperative validations.

Check out complete details of these validation types at [JSF Validation Example](#).

20. What are different types of expressions supported by JSF EL?

JSF Expression Language supports following types of expressions.

1. Immediate value expressions
2. Deferred value expressions

3. Value expression and method expression

Check out complete details about these at [JSF EL Tutorial](#).

21. What are immediate and deferred value expressions?

Immediate expressions are evaluated and results are rendered as soon as the page is displayed initially. The syntax for immediate evaluation is `${}`.

Deferred expressions are evaluated during the lifecycle phase whenever it is requested by the user. The syntax for deferred evaluation is `#{expression}`.

22. Explain value expression and method expressions?

Value expressions usually fetch a value or set a value. These expressions can be further categorized into rvalue and lvalue expressions. lvalue expressions can both read and write data whereas rvalue expressions can only read data.

A method expression allows user to invoke a public method of the bean that returns the result necessary for validating the data component and handling events.

23. Explain @ViewScoped, @SessionScoped, @CustomScoped and @RequestScoped annotations?

@ViewScoped: annotation indicates that the bean is alive as long as the user interacts with the same JSF view page in the browser.

@SessionScoped: annotation indicates that the bean is valid as long as the HTTP session is alive.

@CustomScoped: annotation indicates that the bean lives as long as the bean's entry in the custom Map which is created for this scope lives.

@RequestScoped: annotation indicates that the Bean lives as long as the HTTP request-response lives.

24. Explain different ways of declaring a managed bean in JSF?

1. Use `@ManagedBean` annotation in the java class indicating that the class is a managed bean as;

```
@ManagedBean(name="Greetings", eager="true")
```

If the name attribute is not specified the name is defaulted to the class name as java naming standards. For example class Car will be named "car" and CarDetails will be named "carDetails".

2. Declare the managed bean in faces-config.xml file as;

```
<managed-bean>  
    <managed-bean-name>Greetings</managed-bean-name>  
    <managed-bean-class>com.Greetings.Greetings</managed-bean-class>
```



```
<managed-bean-scope>request</managed-bean-scope>  
</managed-bean>
```

25. What is the significance of name and eager attributes in managed bean?

name: The name attribute indicates the managed bean with the name specified. If the name is not specified then the bean name is same as the class name.

eager: If eager is set to "true" then managed bean is created before it is requested for the first time and if set to false the bean is created when it is requested.

26. Mention some of the validator tags used in JSF?

- `f:validateLength`: Validates length of a string
- `f:validateLongRange`: Validates range of numeric value
- `f:validateDoubleRange`: Validates range of float value
- `f:validateRegex`: Validate JSF component with a given regular expression

27. What are the benefits of using JSF Framework?

Some of the benefits of using JSF framework are;

- Clean separation between presentation and business logic.
- Manages UI state across multiple server requests.
- Implementation of custom components.
- Easier flow of data between the components.
- JSF specs that helps custom implementations such as PrimeFaces

28. What are different JSF Converter tags?

- `f:convertNumber`: tag is used to convert a string value to a number of required format.
- `f:convertDateTime`: tag is used to convert a string value to a date of required format.
- `CustomConverter`: allows user to define their own convertor in JSF.

29. List the benefits of Expression Language?

- Arithmetic, logical, relational operations can be used in expression language.
- Automatic type conversion.
- Shows missing values as empty strings instead of `NullPointerException`.
- Provides easy access to predefined objects such as `request`.

30. What is a backing bean?

A JavaServer Faces application includes one or more backing beans, each of which is a type of managed bean that can be associated with the components used in a particular page.

31. What are standard JSF tag libraries?

1. JSF Core Tags library
2. JSF HTML tags library

Use below namespace configurations to use them in JSF xhtml pages.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets"
      xmlns:c="http://java.sun.com/jsf/core">
```

The html tags can now be used as with the h prefix as <h:head>,<h:form> etc and core tags with c prefix as <c:validateBean>,<c:validator> etc.

32. Mention some of the functions that the backing bean method performs?

- Validating a component's data
- Handling an event fired by a component
- Performs processing to determine the next page to which the application must navigate

33. What are different implementations of JSF API?

- ADF Faces: Oracle's implementation for the JSF standard.
- Reference Implementation (RI): by Sun Microsystems.
- Apache MyFaces: open source JavaServer Faces (JSF) implementation.
- Primefaces: JSF components with Ajax framework.

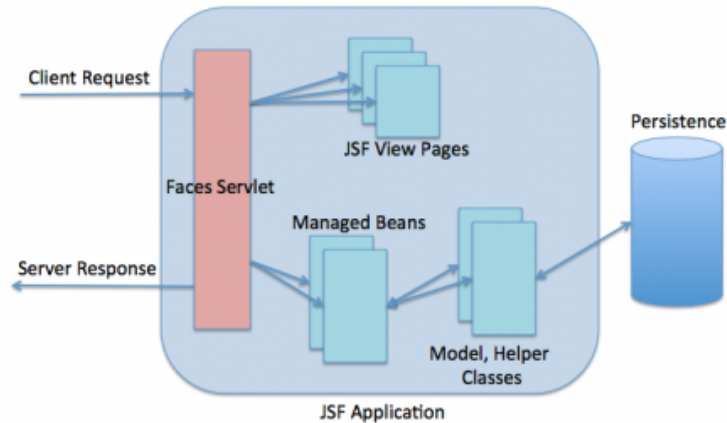
34. Explain JSF architecture?

JSF is developed based on the Model View Controller(MVC) pattern. The Model View Controller separates the business logic from presentation.

The JSF application contains

- UI components represented as stateful objects on the server
- Server-side helper classes
- Validators, event handlers, and navigation handlers
- Application configuration resource file for configuring application resources
- JavaBeans components as models containing application-specific functionality and data
- A custom tag library for representing event handlers and validators
- A custom tag library for rendering UI components

Below image shows the JSF applications architecture diagram.



35. How different components are rendered in JSF page?

JSF components are rendered in the xhtml pages by the tag libraries included, such as JSF core, html and facelets tag libraries.

36. Can the JSF support multiple faces configuration files?

Yes, any number of faces configuration files can be used but should be declared in the web.xml file as shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
  <context-param>
    <param-name>javax.faces.CONFIG_FILES</param-name>
    <param-value>/WEB-INF/faces-config1.xml,/WEB-INF/faces-config2.xml</param-
value>
  </context-param>
</web-app>
```

37. What are the differences between a Backing Bean and Managed Bean?

Backing Beans should be defined in request scope whereas managed bean can be defined in request, session or application scopes. Backing bean is referenced by a form whereas a managed bean is registered with JSF and created automatically when needed.

38. How to display error messages in JSF?

The `h:messages` tag shows all the error messages at one place corresponding to UI elements. The "for" attribute can be used to represent the field for which error message has to be displayed.

For a complete example, please read [JSF Error Messages Example](#).

39. What is the significance of `selectOne` menu tag in JSF?

The `selectOne` enables users to select a single value from the list of values. This component can be rendered as a list box, a set of radio buttons or a menu.

40. Explain immediate and rendered attributes?

The immediate attribute if set to true can force validations, events and conversions processed during request phase of the lifecycle. Command component's immediate attribute indicates what happens when the component gets activated. If the button's immediate attribute is set to true and associated text field's immediate attribute set to false then the event is processed without applying the field's value to the model. In other words the value entered in the field does not even reach the model when the button is clicked but immediately processed in the above scenario.

The rendered attribute indicates whether a component should be rendered or not in the view page. Rendered attribute can use arithmetic operators and literals with `rvalue` expression but not `lvalue` expressions.

41. Mention two ways of binding supported by JSF?

1. Binding the component's value to a bean property or other external data source
2. Binding the component's instance to a bean property

42. What are required configurations for JSF framework?

There are two configuration files namely;

1. `web.xml`: This is the general web application configuration file containing the details of deployment. This contains the `faces-config` file responsible for handling the application.
2. `faces-config.xml`: allows to configure the application, managed beans, converters, validators, and navigation.

43. What is JSF Navigation Rule?

The rules provided by JSF Framework to describe the view to be shown when a button or link is clicked is called a navigation rule. For complete details, please read [JSF Navigation Rules Tutorial](#).

44. What is the role of JSF converter tags?

The convertor tags convert the UI component's data to object used in a managed bean and vice versa. For example, these tags can convert a text into date object and can validate the format of input as well.

45. **List the benefits of data table tags in JSF?**

- DataTable can iterate over collection or array of values to display data.
- DataTable provides attributes to modify its data in easy way

Read more at [JSF Data Table Example](#).

46. **How to implement internationalization (i18n) in JSF?**

Internationalization is a mechanism in which status messages, GUI component labels, currency, date are stored outside the source code in resource bundles and retrieved dynamically rather than hardcoding in the program based on the user locale.

47. **What is component rendering model?**

The JavaServer Faces component architecture is designed such that the functionality of the components is defined by the component classes, whereas the actualities of the rendering itself can be defined by a separate renderer called as component rendering model.

48. **What is a render kit?**

A render kit defines how component classes map to component tags that are appropriate for a particular client.

49. **What is view object?**

A view object is a model object used specifically in the presentation tier but defined outside it. It contains the data that must be displayed in the view layer and the logic to validate user input, handle events, and interact with the business-logic tier.

50. **What do you mean by Bean Scope?**

Bean scope is the mechanism of binding the beans and other objects to be available in the different components of a web application.

51. **What is the difference between JSF-1 and JSF-2?**

- JSF1 does not support the usage of annotations but JSF2 supports annotations that makes it far better than JSF1.
- JSF1 Ajax support was limited, however JSF2 support all the ajax features.
- JSF1 does not provide Template concepts whereas JSF2 does.
- JSF1 was first released and API was having a lot of bugs, JSF2 is totally redesigned and can be used as MVC framework in larger applications.

52. **Can we have a JSF application without faces-config.xml?**

In JSF 1.0 faces-config.xml is mandatory whereas JSF 2.0 supports annotations which does not need faces-config.xml.

53. What are some of the best practices for JSF application?

The best practices for JSF application includes

- Avoid using JSF components for static value.
- Short component Id
- Avoid component bindings
- Facelets for dynamic includes

54. How do you compare JSF with Spring Framework?

Spring uses Inversion of Control and **Dependency Injection** whereas JSF does not.

Spring has built in modules for Login-Logout available for ready integration whereas in JSF we have to write the login feature manually.

Since Spring uses dependency injection the user based pojo classes can be injected with springs whereas JSF is tightly coupled with Java EE architecture.

55. Why JSF is not so popular as MVC framework like Spring MVC, even though its older?

Some of the reasons I could thought of are;

- There are many implementations of JSF like Mojarra, Primefaces, Richfaces etc which makes it difficult for the user to comprehend learn and adapt to whereas Spring MVC has only one implementation maintained by a Single group of developers which avoids confusion.
- Spring has got great integration with data management through standalone or ORM frameworks out of the box which is lacking in JSF based implementation.
- DI and IOC **design patterns** makes it very easy to integrate existing legacy applications with new Spring based applications whereas JSF does not have such kind of capabilities.
- JSF is a component based framework whereas Spring is a Request-Response based framework and hence easy to understand and relates closely to MVC, Struts2 and other similar frameworks

56. Can we integrate JSF with other popular frameworks such as Spring, Hibernate etc?

Yes, we can integrate JSF framework with Spring, Hibernate, JDBC etc. Since JSF is more focused on view components, we can utilize it for user interface and other frameworks as backend server side integration and ORM tools. Some of the posts you should look for integrations are;

- **JSF and JDBC integration example**

- [JSF and Spring MVC integration example](#)
- [JSF, Spring MVC and Hibernate integration example](#)
- [JSF Authentication Login Logout example](#) showing use of HttpSession for session management

That's all for JSF Interview Questions and Answers, I hope they will help you in interviews for JSF related questions.

FILED UNDER: [INTERVIEW QUESTIONS](#), [JSF](#)

About Pankaj

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on [Google Plus](#), [Facebook](#) or [Twitter](#). I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on [Youtube](#).

[« JSF Authentication Login Logout Database Example](#)

[JSF Tutorial »](#)

Comments

Prasad says

OCTOBER 20, 2016 AT 9:24 PM

Hi pankaj i'm going to attend interview on jsf with 2 yrs experience, I preparing these questions is it enough or not?

[Reply](#)

Big B says

OCTOBER 19, 2016 AT 4:03 AM

Great! Great!! Great!!! Great!!!!

[Reply](#)

Asuvani says

JANUARY 12, 2016 AT 6:20 PM

Hi,

Without Face-Config.xml file JSF2.0 will not work?

[Reply](#)

Uwe Nachname says

JULY 18, 2016 AT 9:08 AM

I think that JSF2 will work without faces-config.xml because of annotation, won't it ?

[Reply](#)

Pankaj says

JULY 18, 2016 AT 11:46 AM

Yes you are right.

[Reply](#)

Polyk says

OCTOBER 29, 2015 AT 2:40 AM

PrimeFaces and RichFaces are not JSF implementattions, they are components librairies used by JSF

[Reply](#)

Thien says

OCTOBER 19, 2015 AT 5:10 PM

Hi Pankaj Kumar. I have a question. I can use method in class DefaultSubMenu the same method setCommand of class DefaultMenuItem. Because i want to get id of button SubMenu when i clicked it and SubMenu don't have method allow access event if i clicked it. Thank advance.

[Reply](#)

Abhijeet Kharkar says

MAY 21, 2015 AT 11:45 PM

One question which was asked in one of the interviews I attended:

In which of the 6 phases of JSF will a FILTER used in JSF fall??

Can u explain a bit on this?

[Reply](#)

Thuận says

JUNE 1, 2016 AT 8:53 AM

Not sure if I understand correctly your question, however base 2 these references: filter is always executed before any faces in jsf.

http://docs.oracle.com/cd/B32110_01/web.1013/b28959/filters.htm

<http://stackoverflow.com/questions/9603853/how-to-run-a-phaselistener-before-all-filters>

[Reply](#)

Dave says

MARCH 15, 2015 AT 11:04 AM

These are awesome questions and answers, will help me a lot in coming interview. Thanks...

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

☐

Save my name, email, and website in this browser for the next time I comment.

DOWNLOAD ANDROID APP



IMPORTANT INTERVIEW QUESTIONS

Java Interview Questions

- > [Core Java Interview Questions](#)
- > [String Interview Questions](#)
- > [Multithreading Interview Questions](#)
- > [Collections Interview Questions](#)
- > [Exception Interview Questions](#)
- > [Java Programming Interview Questions](#)
- > [Java 8 Interview Questions Part 1](#)
- > [Java 8 Interview Questions Part 2](#)
- > [Servlet Interview Questions](#)

- > [JSP Interview Questions](#)
- > [Struts 2 Interview Questions](#)
- > [JDBC Interview Questions](#)
- > [Spring Interview Questions](#)
- > [Hibernate Interview Questions](#)
- > [JSF Interview Questions](#)
- > [Web Services Interview Questions](#)
- > [Scala Basic Interview Questions](#)
- > [Scala Intermediate Interview Questions](#)
- > [Scala Advanced Interview Questions](#)
- > [Scala Interview Questions Summary](#)
- > [Common Job Interview Questions](#)

Miscellaneous

- > [Java ClassLoader](#)
- > [String StringBuffer StringBuilder](#)
- > [Java is Pass By Value](#)
- > [Java Heap vs Stack Memory](#)

RECOMMENDED TUTORIALS

Java Tutorials

- > [Java IO](#)
- > [Java Regular Expressions](#)
- > [Multithreading in Java](#)
- > [Java Logging](#)
- > [Java Annotations](#)
- > [Java XML](#)
- > [Collections in Java](#)
- > [Java Generics](#)
- > [Exception Handling in Java](#)
- > [Java Reflection](#)
- > [Java Design Patterns](#)
- > [JDBC Tutorial](#)

Java EE Tutorials

- > [Servlet JSP Tutorial](#)
- > [Struts2 Tutorial](#)
- > [Spring Tutorial](#)
- > [Hibernate Tutorial](#)
- > [Primefaces Tutorial](#)
- > [Apache Axis 2](#)
- > [JAX-RS](#)
- > [Memcached Tutorial](#)

