

## Descriptive question on JSP

### 1. What do you understand by JSP Actions?

JSP actions are tags that direct the server to use existing components or control the behavior of the JSP engine. JSP Actions consist of a typical (XML-based) prefix of "jsp" followed by a colon, followed by the action name followed by one or more attribute parameters.

There are six JSP Actions:

```
<jsp : include / >  
<jsp : forward / >  
<jsp : plugin / >  
<jsp : useBean / >  
<jsp : setProperty / >  
<jsp : getProperty / >
```

### 2. What is the difference between `<jsp:forward page = ... >` and `response.sendRedirect(url)`?

- The element forwards the request object containing the client request information from one JSP file to another file. The target file can be an HTML file, another JSP file, or a servlet.
- `sendRedirect` sends HTTP temporary redirect response to the browser, and browser creates a new request to go the redirected page. The `response.sendRedirect` also kills the session variables.

### 3. Identify the advantages of JSP over Servlet?

JSP can contain HTML, JavaScript, XML and Java Code whereas Servlets can contain only Java Code, making JSPs more flexible and powerful than Servlets.

However, Servlets have their own place in a J2EE application and cannot be ignored altogether. They have their strengths too which cannot be overseen.

### 4. What are all the different scope values for the `<jsp:useBean>` tag?

`<jsp : useBean >` tag is used to use any java object in the jsp page. Here are the scope values for `<jsp : useBean >` tag:

- a) page
- b) request
- c) session and
- d) application

### 5. What is JSP Scriptlet?

JSP Scriptlets is a term used to refer to pieces of Java code that can be embedded in a JSP Page. Scriptlets begins with `<%` tag and ends with `%>` tag. Java code written inside scriptlet executes every time the JSP is invoked.

### 6. What you will handle the runtime exception in your jsp page?

The `errorPage` attribute of the page directive can be used to catch run-time exceptions automatically and then forwarded to an error processing page. You can define the error page to which you want the request forwarded to, in case of an exception, in each JSP Page.

### 7. What are the implicit objects in JSP?

Implicit objects are the objects available to the JSP page. These objects are created by Web container and contain information related to a particular request, page, or application. The JSP implicit objects are:

`application, config, exception, out, page, pageContext, request, response` and `session`

**8. Why we use Servlets?**

Servlets are used to process the client requests.

- \* A Servlet can handle multiple requests concurrently and be used to develop high performance of system

- \* A Servlet can be used to load balance among several servers, as Servlet can easily forward request.

**9. Write the syntax of EL expression? Why we use them?**

-Syntax of EL Expression:

EL expressions are always within the braces { ..... } and prefixed with the \$ or, # sign as:

```
<jsp:useBean id="bean" class="class"/>
```

```
${bean.name}
```

-Expression language use to easily access application data stored in JavaBeans components. For example, the JSP expression language allows a page author to access a bean using simple syntax such as `${name}` for a simple variable or `${name.foo.bar}` for a nested property.

**10. What is Java Server Pages Standard Tag Library (JSTL)?**

Java Server Pages Standard Tag Library (JSTL) encapsulates as simple tags the core functionality common to many Web applications. It also provides a framework for integrating existing custom tags with JSTL tags.

It is a collection of four tag library. They are –

- (a)Core (b) Internationalization (118n) and formatting

- (c) Relational database access (d) XML processing

**11. What is deployment descriptor?**

The deployment descriptor(web.xml) is an xml file that contains the basic and most important information that is required to deploy a web application (Servlet)

Without this, the web server would not know, which requests to entertain/consider as requests to access this servlet.

**12. What is Tag Library Descriptor (TLD)?**

A tag library descriptor is an XML document that contains information about a library as a whole and about each tag contained in the library. TLDs are used by a web container to validate the tags and by JSP page development tools. A TLD must begin with a root `taglib` element.

The syntax for the taglib directive is as follows:

```
<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
```

**13. Difference between GET and POST in Java Servlet?**

Get sends information from the browser to the Servlet as contents appended to the query string in the URL while Post uses hidden variables

- \* Because of the above point, post is a lot safer than get

- \* Get has a size limitation - i.e., we can send only approximately 1 Kb of data while Post can send a significantly higher amount of data

- \* Get is the most common type of sending data from a browser to a servlet, while Post is gaining popularity because of its size and safety which is better than the get.

#### 14. Write the Expanded Directory Format?

The web application in its predefined structure is simply copied into the container's deployment directory.

Root-web-context

META-INF

Context.xml

WEB-INF

Web.xml

Lib-all libraries

Classes-all package and classes

Index.jsp and other jsp pages

#### 15. What is Taglib? Write the syntax of taglib?

A tag library (commonly known as a taglib) is simply a collection of one or more custom tags that are generally related in some way. For example the JSP 2.0 specification introduced a standard tag library known as the JSTL.

For example, suppose the **custlib** tag library contains a tag called **hello**. Your JSP file as follows:

```
<%@ taglib uri="http://www.example.com/custlib" prefix="mytag" %>
<html>
<body><mytag:hello/></body></html>
```

#### 16. How do you configure a Servlet?

First you configure the servlet. This is done using the <servlet> element. Here you give the servlet a name, and writes the class name of the servlet. Second, you map the servlet to a URL or URL pattern. This is done in the <servlet-mapping> element. In the above example, all URL's ending in html are sent to the servlet.

```
<servlet>
<servlet-name>controlServlet</servlet-name>
<servlet-class>org.idb.j2ee.ControlServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>controlServlet</servlet-name>
<url-pattern>*.html</url-pattern>
</servlet-mapping>
```

#### 17. Write the default value of EL expression.

Default values are type-correct values that are assigned to a sub-expression when there is a problem and errors are exceptions to be thrown. An example of such a default value is 'infinity'. This value is assigned to an expression that results in a divide by zero.

Exp: \${2/0}