**JSP SETUP( example server JBoss)**

**Prerequisites:**

1. JSP requires JDK 1.7 or higher.
2. Java 7 or higher

**Installation:**

1. Download JBoss Wildfly 8.

**JSP Actions:**

Jsp actions used to control the behavior of the servlet engine. You can dynamically insert a file, forward the user to another page.

**Attributes:**

There are two attributes that are all common to all elements.

**Id attribute:**

The id attribute uniquely identifies the Action element, and allows the action to be referenced inside the JSP page. If the Action creates an instance of an object the id value can be used to reference it through the implicit object PageContext

**Scope attribute:**

This attribute identifies the lifecycle of the Action element. The id attribute and the scope attribute are directly related, as the scope attribute determines the lifespan of the object associated with the id. The scope attribute has four possible values: (a) page, (b)request, (c)session, and (d) application.

**Reference:**

#### <http://www.tutorialspoint.com/jsp>

**JSP implicit objects:**

These are created by container while translating JSP page to Servlet source to help developers. We can use these objects directly in **scriptlets** that goes in service method, however we can’t use them in JSP Declaration because that code will go at class level.

We have 9 implicit objects that we can directly use in JSP page. Seven of them are declared as local variable at the start of *\_jspService()* method whereas two of them are part of *\_jspService()* method argument that we can use.

1. [out Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-out-object)
2. [request Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-request-object)
3. [response Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-response-object)
4. [config Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-config-object)
5. [application Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-application-object)
6. [session Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-session-object)
7. [pageContext Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-pageContext-object)
8. [page Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-page-object)
9. [exception Object](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-exception-object)
10. [JSP Implicit Objects Example](http://www.journaldev.com/2038/jsp-implicit-objects-with-examples#jsp-implicit-objects-example)
11. **out Object:**

JSP out implicit object is instance of javax.servlet.jsp.JspWriter implementation and it’s used to output content to be sent in client response. This is one of the most used JSP implicit object and thats why we have JSP Expression to easily invoke *out.print()*method.

1. **request Object**

JSP request implicit object is instance of javax.servlet.http.HttpServletRequestimplementation and it’s one of the argument of JSP service method. We can use request object to get the request parameters, cookies, request attributes, session, header information and other details about client request.

1. **response Object**

JSP response implicit object is instance of javax.servlet.http.HttpServletResponseimplementation and comes as argument of service method. We can response object to set content type, character encoding, header information in response, adding cookies to response and redirecting the request to other resource.

1. **config Object**

JSP config implicit object is instance of javax.servlet.ServletConfig implementation and used to get the JSP init params configured in deployment descriptor.

1. **application Object**

JSP application implicit object is instance of javax.servlet.ServletContextimplementation and it’s used to get the context information and attributes in JSP. We can use it to get the RequestDispatcher object in JSP to forward the request to another resource or to include the response from another resource in the JSP.

1. **session Object**

JSP session implicit object is instance of javax.servlet.http.HttpSessionimplementation. Whenever we request a JSP page, container automatically creates a session for the JSP in the service method.

Since session management is heavy process, so if we don’t want session to be created for JSP, we can use page directive to not create the session for JSP using <%@ page session="false" %>. This is very helpful when our login page or the index page is a JSP page and we don’t need any user session there.

1. **pageContext Object**

JSP pageContext implicit object is instance of javax.servlet.jsp.PageContext [abstract class](http://www.journaldev.com/1582/abstract-class-in-java-with-example) implementation. We can use pageContext to get and set attributes with different scopes and to forward request to other resources. pageContext object also hold reference to other implicit object.

1. **page Object**

JSP page implicit object is instance of java.lang.Object class and represents the current JSP page. page object provide reference to the generated servlet class. This object is very rarely used.

1. **exception Object**

JSP exception implicit object is instance of java.lang.Throwable class and used to provide exception details in JSP error pages. We can’t use this object in normal JSP pages and it’s available only in JSP error pages.

1. **JSP Implicit Objects Example**

Here is a simple JSP page that’s showing usage of JSP implicit objects. I am not providing exception object example, that will be covered in JSP Exception Handling post.

**Reference:**

#### <http://www.journaldev.com/2038/jsp-implicit-objects-with-examples>

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Create a *Dynamic Web Project* with a name *HelloWeb* and create a package*com.tutorialspoint* under the *src* folder in the created project. |
| 2 | Drag and drop below mentioned Spring and other libraries into the folder *WebContent/WEB-INF/lib*. |
| 3 | Create a Java classes *Student* and *StudentController* under the *com.tutorialspoint* package. |
| 4 | Create Spring configuration files *Web.xml* and *HelloWeb-servlet.xml* under the*WebContent/WEB-INF* folder. |
| 5 | Create a sub-folder with a name *jsp* under the *WebContent/WEB-INF* folder. Create a view files *student.jsp* and *result.jsp* under this sub-folder. |
| 6 | The final step is to create the content of all the source and configuration files and export the application as explained below. |

### Note: Save the jsp file outside the WEB-INF folder. Change the web-servlet.xml corresponding to that.