**Servlet – Session Tracking**

Servlets are the Java programs that run on the Java-enabled web server or application server. They are used to handle the request obtained from the webserver, process the request, produce the response, then send a response back to the webserver

HTTP is a “stateless” protocol, which means that each time a client requests a Web page, the client establishes a new connection with the Web server, and the server does not retain track of prior requests.

* The conversion of a user over a period of time is referred to as a **session**. In general, it refers to a certain period of time.
* The recording of the object in session is known as**tracking**.
* **Session tracking** is the process of remembering and documenting customer conversions over time. Session management is another name for it.
* The term “**stateful web application**” refers to a web application that is capable of remembering and recording client conversions over time.

**Why is Session Tracking Required?**

* Because the HTTP protocol is stateless, we require Session Tracking to make the client-server relationship stateful.
* Session tracking is important for tracking conversions in online shopping, mailing applications, and E-Commerce applications.
* The HTTP protocol is stateless, which implies that each request is treated as a new one. As you can see in the image below.

**Deleting Session Data**

We have numerous alternatives once you’ve finished processing a user’s session data.

1. Remove a specific attribute You can delete the value associated with a specific key by calling the public void removeAttribute(String name) function.
2. Delete your whole session. To delete an entire session, use the public void invalidate() function.
3. Setting Session Timeout You may set the timeout for a session separately by calling the public void setMaxInactiveInterval(int interval) function.
4. Log the user out On servers that support servlets 2.4, you may use the logout method to log the client out of the Web server and invalidate all of the users’ sessions.
5. web.xml Configuration If you’re using Tomcat, you may set the session timeout in the web.xml file, in addition to the ways listed above.

**<session-config>**

**<session-timeout>20</session-timeout>**

**</session-config>**

The timeout is specified in minutes and overrides Tomcat’s default timeout of 30 minutes.

In a servlet, the getMaxInactiveInterval() function delivers the session’s timeout period in seconds. GetMaxInactiveInterval() returns 900 if your session is set to 20 minutes in web.xml.

**Session Tracking employs Four Different techniques**

1. Cookies
2. Hidden Form Field
3. URL Rewriting
4. HttpSession

**A.**Cookies

Cookies are little pieces of data delivered by the web server in the response header and kept by the browser. Each web client can be assigned a unique session ID by a web server. Cookies are used to keep the session going. Cookies can be turned off by the client.

**B.**Hidden Form Field

The information is inserted into the web pages via the hidden form field, which is then transferred to the server. These fields are hidden from the user’s view.

Illustration**:**

<input type = hidden' name = 'session' value = '12345' >

**C.**URL Rewriting

With each request and return, append some more data via URL as request parameters. URL rewriting is a better technique to keep session management and browser operations in sync.

**D.**HttpSession

A user session is represented by the HttpSession object. A session is established between an HTTP client and an HTTP server using the HttpSession interface. A user session is a collection of data about a user that spans many HTTP requests.

**Illustration:**

HttpSession session = request.getSession( );

Session.setAttribute("username", "password");

The request must be made. Before sending any document content to the client, you must first call getSession(). The following is a list of the most significant methods provided by the

## What is JSP Implicit object?

* JSP implicit objects are created during the translation phase of [JSP](https://www.guru99.com/jsp-tutorial.html) to the servlet.
* These objects can be directly used in scriplets that goes in the service method.
* They are created by the container automatically, and they can be accessed using objects.

## How many Implicit Objects are available in JSP?

There are 9 types of implicit objects available in the container:

1. Out

2. Request

3. Response

4. Config

5. Application

6. Session

7. PageContext

8. Page

9. Exception

Lets study One By One

## Out

* Out is one of the implicit objects to write the data to the buffer and send output to the client in response
* Out object allows us to access the servlet’s output stream
* Out is object of javax.servlet.jsp.jspWriter class
* While working with [servlet](https://www.guru99.com/difference-between-servlets-vs-jsp.html), we need printwriter object

**Example:**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Implicit - JSP1</title>

</head>

<body>

<% int num1=10;int num2=20;

out.println("num1 is " +num1);

out.println("num2 is "+num2);

%>

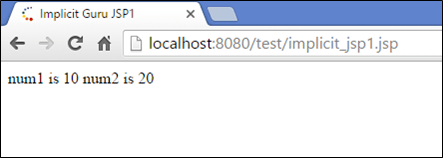
</body>

</html>

**Explanation of the code:**

**Code Line 11-12**– out is used to print into output stream

When we execute the above code, we get the following output:



* In the output, we get the values of num1 and num2

## Request

* The request object is an instance of java.servlet.http.HttpServletRequest and it is one of the argument of service method
* It will be created by container for every request.
* It will be used to request the information like parameter, header information , server name, etc.
* It uses getParameter() to access the request parameter.

**Example:**

**<%@ page language="java" contentType="text/html; charset=ISO-8859-1"**

**pageEncoding="ISO-8859-1"%>**

**<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">**

**<html>**

**<head>**

**<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">**

**<title>Implicit - form JSP2</title>**

**</head>**

**<body>**

**<form action="-.jsp">**

**<input type="text" name="username">**

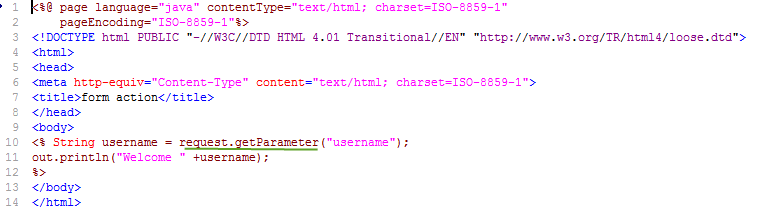
**<input type="submit" value="submit">**

**</form>**

**</body>**

**</html>**

-.jsp (where the action is taken)



**Explanation of code:**

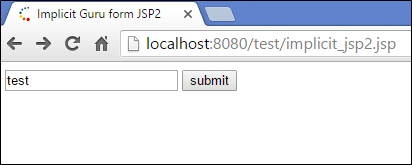
**Code Line 10-13 :** In implicit\_jsp2.jsp(form) request is sent, hence the variable username is processed and sent to -.jsp which is action of JSP.

-.jsp

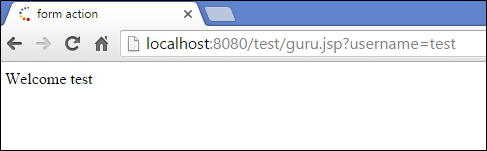
**Code Line10-11:** It is action jsp where the request is processed, and username is taken from form jsp.

When you execute the above code, you get the following output

**Output:**



When you write test and click on the submit button, then you get the following output “Welcome Test.”



## Response

* “Response” is an instance of class which implements HttpServletResponse interface
* Container generates this object and passes to \_jspservice() method as parameter
* “Response object” will be created by the container for each request.
* It represents the response that can be given to the client
* The response implicit object is used to content type, add cookie and redirect to response page

**Example:**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Implicit - JSP4</title>

</head>

<body>

<%response.setContentType("text/html"); %>

</body>

</html>

# JSP Elements – Declaration, Syntax & Expression

In this tutorial, we will be learning the basic tags of JSP and how to add comments into JSP. Along with this, we will also create a JSP and run that JSP on the server.

## JSP Declaration

* A declaration tag is a piece of[Java](https://www.guru99.com/java-tutorial.html)code for declaring variables, methods and classes. If we declare a variable or method inside declaration tag it means that the declaration is made inside the servlet class but outside the service method.
* We can declare a static member, an instance variable (can declare a number or string) and methods inside the declaration tag.

**Syntax of declaration tag:**

<%! Dec var %>

Here Dec var is the method or a variable inside the declaration tag.

**Example:**

In this example, we are going to use the declaration tags

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>- Declaration Tag</title>

</head>

<body>

<%! int count =10; %>

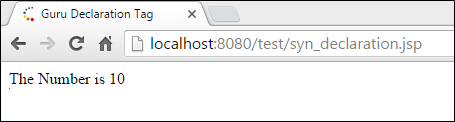
<% out.println("The Number is " +count); %>

</body>

</html>

**Explanation the code:**

**Code Line 10:** Here we are using declaration tag for initializing a variable count to 10.



**Output:**

The variable which is declared in the declaration tag is printed as output.

## JSP Scriptlet

* Scriptlet tag allows to write Java code into JSP file.
* JSP container moves statements in \_jspservice() method while generating servlet from jsp.
* For each request of the client, service method of the JSP gets invoked hence the code inside the Scriptlet executes for every request.
* A Scriptlet contains java code that is executed every time JSP is invoked.

**Syntax of Scriptlet tag:**

<% java code %>

Here <%%> tags are scriplets tag and within it, we can place java code.

**Example:**

In this example, we are taking Scriptlet tags which enclose java code.

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>- Scriplet</title>

</head>

<body>

<% int num1=10;

int num2=40;

int num3 = num1+num2;

out.println("Scriplet Number is " +num3);

%>

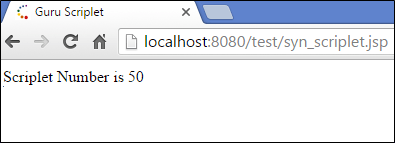
</body>

</html>

**Explanation of the code:**

**Code Line 10-14:**In the Scriptlet tags where we are taking two variables num1 and num2 . Third variable num3 is taken which adds up as num1 and num2.The output is num3.

When you execute the code, you get the following output:



**Output:**

The output for the Scriptlet Number is 50 which is addition of num1 and num2.

## JSP Expression

* Expression tag evaluates the expression placed in it.
* It accesses the data stored in stored application.
* It allows create expressions like arithmetic and logical.
* It produces scriptless JSP page.

**Syntax:**

<%= expression %>

Here the expression is the arithmetic or logical expression.

**Example:**

In this example, we are using expression tag

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>- Expression</title>

</head>

<body>

<% out.println("The expression number is "); %>

<% int num1=10; int num2=10; int num3 = 20; %>

<%= num1\*num2+num3 %>

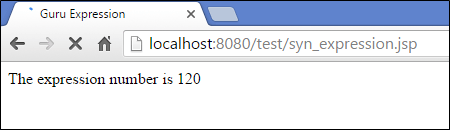
</body>

</html>

**Explanation of the code:**

**Code Line 12:** Here we are using expression tags where we are using an expression by multiplying two numbers i.e. num1 and num 2 and then adding the third number i.e. num3.

When you execute the above code, you get the following output:



**Output:**

The expression number is 120 where we are multiplying two numbers num1 and num2 and adding that number with the third number.

## JSP Comments

Comments are the one when JSP container wants to ignore certain texts and statements.

When we want to hide certain content, then we can add that to the comments section.

Syntax:

<% -- JSP Comments %>

T his tags are used to comment in JSP and ignored by the JSP container.

<!—comment –>

This is HTML comment which is ignored by browser

**Example:**

In this example, we are using JSP comments

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>- Comments</title>

</head>

<body>

<%-- - Comments section --%>

<% out.println("This is comments example"); %>

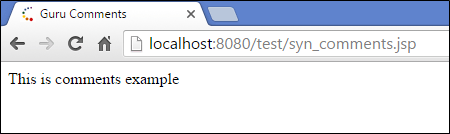
</body>

</html>

**Explanation of the code:**

**Code Line 10:** Here we are adding JSP comments to the code to explain what code has. It is been ignored by the JSP container

When you execute the above code you get the following output:



**Output:**

We get the output that is printed in println method. Comments are ignored by container

## Creating a simple JSP Page

* A JSP page has an HTML body incorporated with Java code into it
* We are creating a simple JSP page which includes declarations, scriplets, expressions, comments tags in it.

**Example:**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>- JSP Example</title>

</head>

<body>

<%-- This is a JSP example with scriplets, comments , expressions --%>

<% out.println("This is - JSP Example"); %>

<% out.println("The number is "); %>

<%! int num12 = 12; int num32 = 12; %>

<%= num12\*num32 %>

Today's date: <%= (new java.util.Date()).toLocaleString()%>

</body>

</html>

**Explanation of the code:**

**Code Line 1:** Here we are using directives like language, contentType and pageEncoding. Language is Java and content type is text/html with standard charset ISO 8859. Page encoding is standard charset.

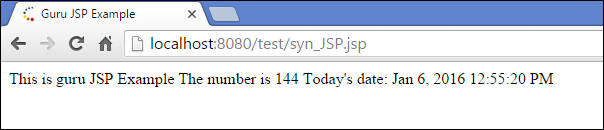
**Code Line 11:** Here we are using JSP comments to add comments to the JSP

**Code Line 14:** Here we are declaring variables num12 and num32 initializing with 12.

**Code Line 15:** Here we are using an expression where we are multiplying two numbers num12 and num32.

**Code Line 16:** Here we are fetching today’s date using date object.

When you execute the above code, you get the following output



**Output:**

We are printing overhere,

* This is - JSP example.
* The number is num12\*num32 (12\*12).
* Today’s date is the current date

## How to run simple JSP Page