JSP Implicit Objects

So far we know that the developer can create javabeans and interact with java objects. But there are several objects that are automatically available in JSP called Implicit Object.

JSP implicit objects are created during the translation phase of JSP to the servlet.

These objects can be directly used in scriplets that goes in the service method.

There are **9 jsp implicit objects**. These objects are *created by the web container* that are available to all the jsp pages.

The available implicit objects are out, request, config, session, application etc.

A list of the 9 implicit objects is given below:

Sr. No	Object	Type
1	out	javax.servlet.jsp.JspWriter
2	request	javax.servlet.http.HttpServletRequest
3	response	javax.servlet.http.HttpServletResponse
4	session	javax.servlet.http.HttpSession
5	config	javax.servlet.http.ServletConfig
6	application	javax.servlet.http.ServletContext
7	exception	java.lang.throwable
8	pageContext	javax.servlet.jsp.PageContext
9	page	java.lang.Object

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, we need PrintWriter object

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.

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

session

- The session is holding "HttpSession" object
- Session object is used to get, set and remove attributes to session scope and also used to get session information

A common example of out, request, response and session implicit object

File name : login.html

```
<html>
   <head>
      <title>User Authentication Page</title>
   </head>
   <body>
      <form action="login_server.jsp" method="get">
```

```
            Enter Username : <input type="text" name="un"><br>
            Enter Password : <input type="password" name="ps"><br>
            <input type="submit" name="log" value="LOGIN">
        </form>
    </body>
</html>
```

File Name : login_server.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Login Server Page</title>
    </head>
    <body>
        <%
            String user=request.getParameter("un");
            String pass=request.getParameter("ps");
            if(user.equals("Admin") && pass.equals("123"))
            {
                session.setAttribute("a", user);
                response.sendRedirect("home.jsp");

            }
            else
            {
                out.println("Invalid username or password!");
            }
        %>
    </body>
</html>
```

File Name : home.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Home Page</title>
    </head>
    <body>
        <h1>Welcome to home page</h1>
        <%
        String s=session.getAttribute("a").toString();
        out.println("Welcome , "+s);
        %>
    </body>
</html>
```

config

In JSP, config is an implicit object of type *ServletConfig*.

This object can be used to get initialization parameter for a particular JSP page.

The config object is created by the web container for each jsp page.

Generally, it is used to get initialization parameter from the web.xml file.

Example :-

File Name : index.html

```
<Html>
<body>
<form action="welcome">
<input type="text" name="uname">
<input type="submit" value="go"><br/>
</form>
</body>
</Html>
```

File Name:- welcome.jsp

```
<Html>
<body>
<%
out.print("Welcome "+request.getParameter("uname"));

String driver=config.getInitParameter("dname");
out.print("Message ="+driver);
%>
</body>
</Html>
```

File Name :- web.xml

```
<web-app>

<servlet>
        <servlet-name>MyServlet</servlet-name>
        <jsp-file>/welcome.jsp</jsp-file>

        <init-param>
                <param-name>dname</param-name>
                <param-value>Example of config implicit object.</param-value>
        </init-param>
</servlet>

<servlet-mapping>
        <servlet-name> MyServlet</servlet-name>
```

```
    <url-pattern>/welcome</url-pattern>
</servlet-mapping>

</web-app>
```

application

In JSP, application is an implicit object of type *ServletContext*.

The instance of ServletContext is created only once by the web container when application or project is deployed on the server.

This object can be used to get initialization parameter from configuaration file (web.xml). It can also be used to get, set or remove attribute from the application scope.

Example:-

File Name : index.html

```
<Html>
<body>
<form action="welcome">
<input type="text" name="uname">
<input type="submit" value="go"><br/>
</form>
</body>
</Html>
```

File Name:- welcome.jsp

```
<Html>
<body>
<%
out.print("Welcome "+request.getParameter("uname"));

String driver=application.getInitParameter("dname");
out.print("Message ="+driver);
%>
</body>
</Html>
```

File Name :- web.xml

```
<web-app>

<servlet>
        <servlet-name> MyServlet</servlet-name>
        <jsp-file>/welcome.jsp</jsp-file>
  </servlet>

<context-param>
        <param-name>dname</param-name>
        <param-value>Example of application implicit object.</param-value>
</context-param>

<servlet-mapping>
        <servlet-name> MyServlet</servlet-name>
        <url-pattern>/welcome</url-pattern>
</servlet-mapping>

</web-app>
```

exception

In JSP, exception is an implicit object of type java.lang.Throwable class. This object can be used to print the exception. But it can only be used in error pages.It is better to learn it after page directive.

Example : -

pageContext

In JSP, pageContext is an implicit object of type PageContext class.The pageContext object can be used to set,get or remove attribute from one of the following scopes:

- page
- request

- session
- application

In JSP, page scope is the default scope.

Example:-

File name : login.html

```
<html>
    <head>
        <title>User Authentication Page</title>
    </head>
    <body>
        <form action="login_server.jsp" method="get">
            Enter Username : <input type="text" name="un"><br>
            Enter Password : <input type="password" name="ps"><br>
            <input type="submit" name="log" value="LOGIN">
        </form>
    </body>
</html>
```

File Name : login_server.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Login Server Page</title>
    </head>
    <body>
        <%
```

```
            String user=request.getParameter("un");
            String pass=request.getParameter("ps");
            if(user.equals("Admin") && pass.equals("123"))
            {

                pageContext.setAttribute("a",user,PageContext.SESSION_SCOPE);
                response.sendRedirect("home.jsp");

            }
            else
            {
                out.println("Invalid username or password!");
            }
        %>
    </body>
</html>
```

File Name : home.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Home Page</title>
    </head>
    <body>
        <h1>Welcome to home page</h1>
        <%
    String s=(String)pageContext.getAttribute("a",PageContext.SESSION_SCOPE);
     out.println("Welcome , "+s);
      %>
   </body>
</html>
```

page

- Page implicit variable holds the currently executed servlet object for the corresponding jsp.
- Acts as **this** object for current jsp page.

Example :-

File Name : page_example.jsp

```
<html>
<head>
<title>Example of page implicit object.</title>
</head>
<body>
<% String pageName = page.toString();
out.println("Page Name is " +pageName);%>
</body>
</html>
```

Expression Language (EL) in JSP

The **Expression Language** (EL) simplifies the accessibility of data stored in the Java Bean component, and other objects like request, session, application etc.

There are many implicit objects, operators and reserve words in EL.

It is the newly added feature in JSP technology version 2.0.

Syntax for Expression Language (EL)

1. ${ expression }

Implicit Objects in Expression Language (EL)

There are many implicit objects in the Expression Language. They are as follows:

Implicit Objects	**Usage**
pageScope	it maps the given attribute name with the value set in the page scope
requestScope	it maps the given attribute name with the value set in the request scope
sessionScope	it maps the given attribute name with the value set in the session scope
applicationScope	it maps the given attribute name with the value set in the application scope

param	it maps the request parameter to the single value
paramValues	it maps the request parameter to an array of values
header	it maps the request header name to the single value
headerValues	it maps the request header name to an array of values
cookie	it maps the given cookie name to the cookie value
initParam	it maps the initialization parameter
pageContext	it provides access to many objects request, session etc.

EL param example

In this example, we have created two files index.jsp and process.jsp. The index.jsp file gets input from the user and sends the request to the process.jsp which in turn prints the name of the user using EL.

index.jsp

```
<form action="process.jsp">
Enter Name:<input type="text" name="name" /><br/><br/>
<input type="submit" value="go"/>
</form>
```

process.jsp

```
Welcome, ${ param.name }
```

download this example

EL sessionScope example

In this example, we printing the data stored in the session scope using EL. For this purpose, we have used sessionScope object.

index.jsp

```
<h3>welcome to index page</h3>
<%
session.setAttribute("user","sonoo");
%>
```

```

<a href="process.jsp">visit</a>
```

process.jsp

```
Value is ${ sessionScope.user }
```

download this example

EL cookie example

index.jsp

```
<h1>First JSP</h1>
<%
Cookie ck=new Cookie("name","abhishek");
response.addCookie(ck);
%>
<a href="process.jsp">click</a>
```

process.jsp

```
Hello, ${cookie.name.value}
```

Precedence of Operators in EL

There are many operators that have been provided in the Expression Language. Their precedence are as follows:

[] .
()
-(unary) not ! empty
* / div % mod
+ - (binary)
< <= > >= lt le gt ge
== != eq ne

&& and
|| or
?:

Reserve words in EL

There are many reserve words in the Expression Language. They are as follows:

lt	le	gt	ge
eq	ne	true	false
and	or	not	instanceof
div	mod	empty	null