

Class 1 (11/11/2024)

Q) Types of scopes in Servlets?

We have three **scopes in Servlets**.

1) Request scope

ex:

HttpServletRequest

2) Session scope

ex:

HttpSession

3) Application scope

ex:

ServletContext

JSP

=====

JSP stands for **Java Server Pages**.

JSP is a **dynamic web resource program** which is **used to develop web applications**.

Limitations with Servlets

=====

- > To work with servlet strong java knowledge is required.
- > It is not suitable for non-java programmers.
- > It does not give implicit objects.
- > Handling exceptions are mandatory.
- > Configuration of each servlet program in web.xml file is mandatory.
- > We can't maintain HTML code and Java code separately.

Advantages of JSP

=====

- > To work with JSP strong java knowledge is not required.
- > It is suitable for java and non-java programmers.

- > It supports tag based language.
- > It allows us to work with custom tags and third party supplied tags.
- > It gives 9 implicit objects.
- > Handling exception is optional.
- > Configuration of each JSP program in web.xml file is optional.
- > We can maintain HTML code and Java code separately.
- > It contains all the features of Servlets.

1.)1st web application development having JSP program as web resource program

Deployment Directory Structure

```

JspApp1
|
|---Java Resources
|
|---WebContent
    |
    |---ABC.jsp
    |
    |---WEB-INF
        |
        |---web.xml
  
```

Note :- In above application we need to add "servlet-api.jar" file in project build path.

ABC.jsp

```

<center>
    <h1>
  
```

Current Date and Time


```

<%
    java.util.Date date=new java.util.Date();
    out.println(date);
%>
  
```

```
</h1>
</center>
```

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <welcome-file-list>
        <welcome-file>ABC.jsp</welcome-file>
    </welcome-file-list>

</web-app>
```

Request url :- http://localhost:2525/JspApp1/

Configuration of JSP program in web.xml

=====

Deployment Directory Structure

```
JspApp1
|
|---Java Resources
|
|---WebContent
    |
    |---ABC.jsp
    |
    |---WEB-INF
        |
        |---web.xml
```

Note :- In above application we need to add "servlet-api.jar" file in project build path.

ABC.jsp

```
<center>
    <h1>
```

Current Date and Time


```

        <%
            java.util.Date date=new java.util.Date();
            out.println(date);
        %>

    </h1>
</center>

```

Web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <servlet>
        <servlet-name>ABC</servlet-name>
        <jsp-file>/ABC.jsp</jsp-file>
    </servlet>

    <servlet-mapping>
        <servlet-name>ABC</servlet-name>
        <url-pattern>/test</url-pattern>
    </servlet-mapping>

</web-app>

```

Request url :- <http://localhost:2525/JspApp1/test>
<http://localhost:2525/JspApp1/ABC.jsp>

How can we access our application only by using url pattern

=====

We can make our application accessible by using url pattern not by using file name. If we place ABC.jsp file inside "WEB-INF" folder.

Deployment Directory Structure

```

JspApp1
|
|---Java Resources
|
|---WebContent

```

```

|
|
|---WEB-INF
      |
      |---ABC.jsp
      |
      |---web.xml

```

Note :- In above application we need to add "servlet-api.jar" file in project build path.

ABC.jsp

```

<center>
    <h1>

        Current Date and Time <br>

    <%
        java.util.Date date=new java.util.Date();
        out.println(date);
    %>

    </h1>
</center>

```

Web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <servlet>
        <servlet-name>ABC</servlet-name>
        <jsp-file>/WEB-INF/ABC.jsp</jsp-file>
    </servlet>
    <servlet-mapping>
        <servlet-name>ABC</servlet-name>
        <url-pattern>/test</url-pattern>
    </servlet-mapping>

</web-app>

```

Request url :- http://localhost:2525/JspApp1/test (valid)
http://localhost:2525/JspApp1/ABC.jsp (Invalid)

Note:

Servlet container is used to execute servlet program.

JSP container is used to execute JSP program.

But JSP container can't execute JSP program directly. It takes the support of servlet container to execute JSP program.

Internally, for every JSP program one JES class will be created.

JES stands for Java Equivalent Servlet class.

JSP life cycle methods

=====

We have three life cycle methods in JSP.

1) _jspInit()

It is used for instantiation event.

This method will execute just before JES class object creation.

2) _jspService()

It is used for request arrival event.

This method will execute when request goes to JSP program.

3) _jspDestroy()

It is used for destruction event.

This method will execute just before JES class object destruction.

Q)_What is the difference between Servlets and JSP ?

Servlets

To work with servlet strong java knowlege is not required.

It is not suitable for non-java programmers.

JSP

To work with JSP strong java knowledge is required.

It is suitable for java and non-java

programmers.

It does not support tags.

It is faster.

It does not give any implicit object.

Handling exception is mandatory.

Configuration of each servlet program
web.xml
in web.xml file is mandatory.

We can't maintain HTML code and java code
separately.

It supports tags.

It is slower.

It gives 9 implicit objects.

Handling exception is optional.

Configuration of each jsp program in
file is optional.

We can maintain HTML code and java code
Separately.

Phases in JSP

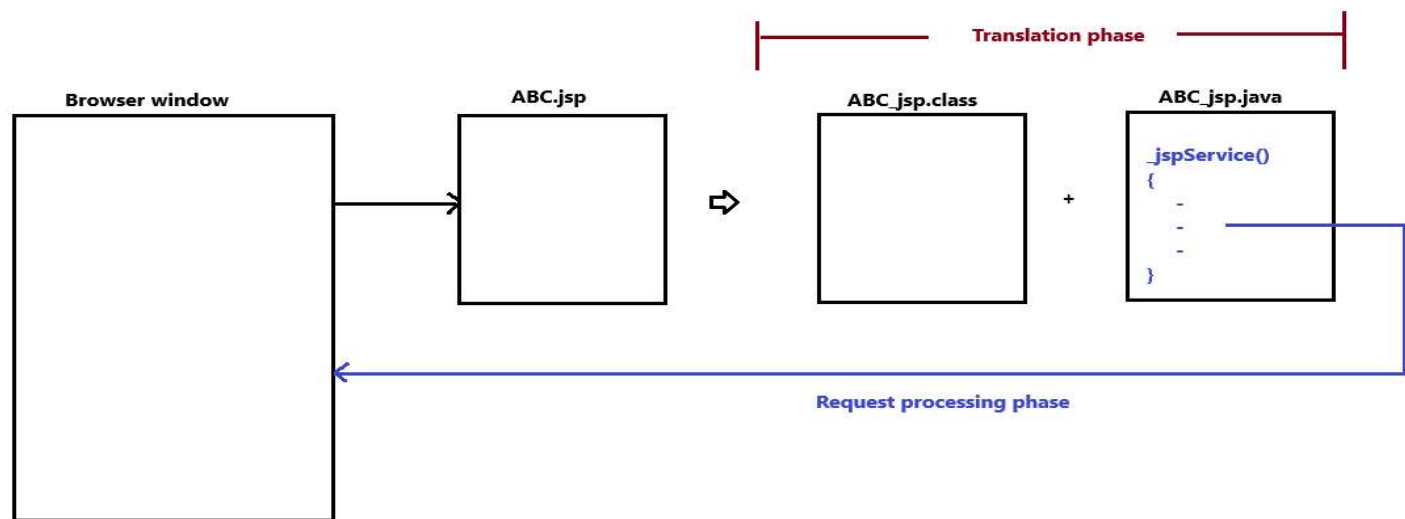
=====

We have two phases in JSP.

1) Translation phase

2) Request Processing phase

Diagram: jsp1.1



1) Translation phase

=====

In translation phase, our **JSP program convert to JES class** (ABC_jsp.class & ABC_jsp.java).

2) Request Processing phase

=====

In request processing phase , our **JES class will be executed and result send to browser window as dynamic response.**

How to enable <load-on-startup> and what happens if we enable <load-on-startup>

we can enable <load-on-startup> inside web.xml file.

Web.xml

```
<web-app>
```

```
  <servlet>
    <servlet-name>ABC</servlet-name>
    <jsp-file>/WEB-INF/ABC.jsp</jsp-file>
    <load-on-startup>1</load-on-startup>
  </servlet>
```

```
  <servlet-mapping>
    <servlet-name>ABC</servlet-name>
    <url-pattern>/test</url-pattern>
  </servlet-mapping>
```

```
</web-app>
```

If we enable <load-on-startup> then our web container performs translation phase during the server startup or during the deployment of web application.

It means JES class object will be created before we give the first request.

Q) What is the difference between HTML and JSP?

HTML

HTML stands for Hypertext Markup Language.

It is used to create static web pages.

We can't create custom tags.

JSP

JSP stands for Java Server Pages.

It is used to create dynamic web pages.

We can create custom tags.

It requires browser window for execution.

It requires server for execution.

It is used to build client side components.

It is used to build server side components.

Interview Question

=====

Q) Write a java program to display the string in a given format?

input:

abc.txt

output:

txt

ex:

```
public class ExampleApp
{
    public static void main(String[] args)
    {
        String fileName="abc.txt";

        int index = fileName.lastIndexOf('.');

        System.out.println(fileName.substring(index+1));
    }
}
```

CLASS 2 (12/11/2024)

Q) **Types of errors in java?**

We have three types of errors in java.

1) Logical Error

2) Compile time Error

3) Runtime Error

JSP Tags/Elements

=====

We have following three tags in JSP.

1) Scripting tags

i) Scriptlet tag

Ex: <% code here %>

ii) Expression tag

Ex: <%= code here %>

iii) Declaration tag

Ex: <%! code here %>

2) Directive tags

i) Page directive tag

Ex: <%@page attribute=value %>

ii) Include directive tag

Ex: <%@include attribute=value %>

3) Action Tags

<jsp:include>

<jsp:forward>

<jsp:useBean>

<jsp:setProperty>

<jsp:getProperty>

and etc.

Comments in JSP :- <%-- comment here --%>

=====

Scriptlet tag :- It is used to **declare java code.**

=====

Syntax :- <% code here %>

Deployment Directory Structure

JspApp2

|

```

|---Java Resources
|
|---WebContent
|
|   |---form.html
|   |---process.jsp
|   |
|   |---WEB-INF
|       |
|       |---web.xml

```

Note :- In above application we need to add "servlet-api.jar" file in project build path.

form.html

```

<form action="process.jsp">

    Name: <input type="text" name="t1"/>

    <input type="submit" value="submit"/>
</form>

```

Web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <welcome-file-list>
        <welcome-file>form.html</welcome-file>
    </welcome-file-list>

</web-app>

```

process.jsp

```

<%
    String name = request.getParameter("t1");

    out.println("Welcome :"+name);
%>

```

Request url :- http://localhost:2525/JspApp2/

ii) Expression tag

=====

The **code** which is **written in a expression tag** will **return to the output stream of a response**.

Hence we don't need to use `out.println()` to print the data in a JSP.

Syntax : - `<%= code here %>`

Note :- Expression tag **does not support semicolon.**

form.html

```
<form action="process.jsp">

    Name: <input type="text" name="t1"/>

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

</form>
```

process.jsp

```
<%
    String name = request.getParameter("t1");
%>
<center>
    <h1>
        <%= "Hey! Welcome : "+name %>
    </h1>
</center>
```

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <welcome-file-list>
        <welcome-file>form.html</welcome-file>
```

```
</welcome-file-list>
```

```
</web-app>
```

Request url :- http://localhost:2525/JspApp2/

iii) Declaration tag

=====

It is used to declare fields and methods.

Syntax: - <%! code here %>

Deployment Directory Structure

```
JspApp3
|
|---Java Resources
|
|---WebContent
    |
    |---process.jsp
    |
    |---WEB-INF
        |
        |---web.xml
```

Note :- In above application we need to add "servlet-api.jar" file in project build path.

process.jsp

```
<%!
    int data=100;

    int cube(int n)
    {
        return n*n*n;
    }
%>

<%= "The value is "+data %> <br>

<%= "The cube of a given number is "+cube(5) %>
```

Web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <welcome-file-list>
        <welcome-file>process.jsp</welcome-file>
    </welcome-file-list>

</web-app>
```

Request url :- http://localhost:2525/JspApp3/

Exception Handling in JSP

=====

Exceptions are **also known as runtime errors**.

Exception may raise any time in our application so handling exceptions is always safer side for the programmer.

There are **two ways to handle the exceptions in jsp**.

- 1) Using errorPage and isErrorPage attribute of page directive tag.
- 2) Using <error-page> element in web.xml file.

1) Using errorPage and isErrorPage attribute of page directive tag

=====

Deployment Directory Structure

```
JspApp4
|
|---Java Resources
|
|---WebContent
    |
    |---form.html
    |---process.jsp
```

```
|---error.jsp
|
|---WEB-INF
|
|---web.xml
```

Note :- In above application we need to add "servlet-api.jar" file in project build path.

form.html

```
<form action="process.jsp">

    No1: <input type="text" name="t1"/> <br>

    No2: <input type="text" name="t2"/> <br>

    <input type="submit" value="divide"/>

</form>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <welcome-file-list>
        <welcome-file>form.html</welcome-file>
    </welcome-file-list>

</web-app>
```

process.jsp

```
<%@page errorPage="error.jsp" %>
<%

    String sno1=request.getParameter("t1");
    String sno2=request.getParameter("t2");

    int a=Integer.parseInt(sno1);
    int b=Integer.parseInt(sno2);
```

```

        int c = a / b;
    %>
<center>
    <h1>
        <%= "Division of two numbers is =" + c %>
    </h1>
</center>

```

error.jsp

```

<%@page isErrorPage="true" %>

<b>
    <i> Sorry! Exception occurred!</i>
</b>
<br>
<%= exception %>

```

Request url : - http://localhost:2525/JspApp4/

2) Using <error-page> element in web.xml file

=====

- This approach is better because we don't need to declare errorPage attribute in each jsp file.
- Defining <error-page> element as a single entry in web.xml file will handle all types of exceptions.

Deployment Directory Structure

```

JspApp4
|
|---Java Resources
|
|---WebContent
    |
    |---form.html
    |---process.jsp
    |---error.jsp
    |
    |---WEB-INF
        |

```


|---web.xml

Note :- In above application we need to add "servlet-api.jar" file in project build path.
form.html

```
<form action="process.jsp">

    No1: <input type="text" name="t1"/> <br>

    No2: <input type="text" name="t2"/> <br>

    <input type="submit" value="divide"/>

</form>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd" id="WebApp_ID" version="3.0">

    <error-page>
        <exception-type>java.lang.Exception</exception-type>
        <location>/error.jsp</location>
    </error-page>

    <welcome-file-list>
        <welcome-file>form.html</welcome-file>
    </welcome-file-list>

</web-app>
```

process.jsp

```
<%
    String sno1=request.getParameter("t1");
    String sno2=request.getParameter("t2");

    int a=Integer.parseInt(sno1);
    int b=Integer.parseInt(sno2);

    int c = a / b;
```

```
%>
<center>
    <h1>
        <%= "Division of two numbers is "+c %>
    </h1>
</center>
```

error.jsp

```
<%@page isErrorPage="true" %>

<b>
    <i> Sorry! Exception occurred!</i>
</b>
<br>
<%= exception %>
```

Request url :- <http://localhost:2525/JspApp4/>

