

Dt : 26/3/2025

1.init():

=>init()-method is used to perform initialization process,which means making the required programming components ready for service()-method.

Method Signature of init():

**public abstract void init(jakarta.servlet.ServletConfig)
throws jakarta.servlet.ServletException;**

2.service():

=>service()-method will perform request handling process,which means accepting the request and providing the response.

Method Signature of service():

**public abstract void service(jakarta.servlet.HttpServletRequest,
jakarta.servlet.HttpServletResponse) throws jakarta.servlet.ServletException,
java.io.IOException;**

3.destroy():

=>destroy()-method is used to close the resources and services opened part of initialization and request handling process.

Method Signature destroy():

public abstract void destroy();

4.getServletInfo():

=>getServletInfo()-method is used to display the servlet-Information.

Method Signature of `getServletInfo()`:

`public abstract java.lang.String getServletInfo();`

5.`getServletConfig()`:

=>`getServletConfig()`-method is used to display the Servlet Configuration details.

Method Signature of `getServletConfig()`:

`public abstract jakarta.servlet.ServletConfig getServletConfig();`

Note:

=>The following three methods are executed automatically in same order and which are known as Servlet-Life-Cycle methods:

(i)`init()`

(ii)`service()`

(iii)`destroy()`

=>In the process of Constructing Servlet-Program,the Servlet-Program must be implemented from 'Servlet-Interface' and the Servlet-Program must implement all abstract methods of 'Servlet-Interface'.

Diagram:

***imp**

Creating and executing Web Application using IDE Eclipse:

Step-1 : Open IDE Eclipse,while Opening select the Workspace and Click 'Launch'

Step-2 : Create 'Dynamic Web Project'

**Click on File->New->Project->Web->select 'Dynamic Web Project' and click 'Next'->
name the project and click 'Finish'**

Step-3 : Add 'servlet-api.jar' file to Project through 'Build path'

**RightClick on Project->Build Path->Configure Build Path->Libraries->select 'Classpath' and
click 'Add External JARs'->Browse and select 'servlet-api.jar' file from "lib" folder of
Tomcat->Open->Apply->Apply and Close.**

Step-4 : Add Web Server(Tomcat) to IDE Eclipse

Click on Servers->Click on 'Click this link to Create new Server'->

Select the Server with version and click 'Next'->

Browse and select 'Tomcat Installation directory'->Click 'Finish'

Step-5 : Construct HTML file to read data to Servlet Program

RightClick on webapp->New->HTML file->name the file and click 'Finish'

user.html

<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

</head>

```
<body>

<form action="dis" method="post">

UserName:<input type="text" name="uname"><br>

MailId:<input type="text" name="mid"><br>

PhoneNo:<input type="text" name="phno"><br>

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

</form>

</body>

</html>
```

Step-6 : Construct 'web.xml' mapping file

RightClick on WEB-INF->New->Other->XML->XML file->name the file as 'web.xml' and click 'Finish'

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<web-app>

  <welcome-file-list>

    <welcome-file>

      user.html

    </welcome-file>

  </welcome-file-list>

</web-app>
```

Step-7 : Create package in 'src/main/java' of 'Java Resources'

Step-8 : Create Servlet-Program in package

DisplayServlet.java

```
package test;

import java.io.*;

import jakarta.servlet.*;

import jakarta.servlet.annotation.*;

@WebServlet("/dis")

public class DisplayServlet implements Servlet
{

    @Override

    public void init(ServletConfig scf) throws ServletException

    {

        //NoCode

    }

    @Override

    public void service(ServletRequest req,ServletResponse res) throws

    ServletException,IOException

    {

        String uName = req.getParameter("uname");
```

```
String mId = req.getParameter("mid");

long phNo = Long.parseLong(req.getParameter("phno"));

PrintWriter pw = res.getWriter();

res.setContentType("text/html");

pw.println("*****User Details*****<br>");

pw.println("UserName:"+uName+"<br>");

pw.println("MailId:"+mId+"<br>");

pw.println("PhoneNo:"+phNo+"<br>");

}

@Override

public void destroy()

{

    //NoCode

}

@Override

public String getServletInfo()

{

    return "Servlet displaying User details....";

}

@Override

public ServletConfig getServletConfig()

{

    return this.getServletConfig();

}
```

}

Step-9 : Execute the Application(Project)

RightClick on Project->Run AS->Run On Server->Select the server and Click 'Finish'

http://localhost:8082/App_Servlet_1/
