

**Dt : 5/2/2025**

**\*imp**

**Servlet API:**

**=>'jakarta.servlet' package is known as Servlet-API, and which provide 'Classes and Interfaces' to develop Servlet-Application.**

**(From Tomcat10-version onwards the Java-Lib is referred by the word 'jakarta' and upto Tomcat9-version Java-Lib is referred by the word 'javax')**

**=>'Servlet' interface from 'jakarta.servlet' package is the root of Servlet-API.**

**=>The following are some important methods of 'Servlet' interface:**

- 1.init()**
- 2.service()**
- 3.destroy()**
- 4.getServletConfig()**
- 5.getServletInfo()**

**1.init():**

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

**Method Signature:**

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

**2.service():**

**=>service()-method is used to accept the request and provide the response.**

**Method Signature:**

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

**3.destroy():**

***=>destroy()-method is used to close the resources and services which are opened while  
init() and service() methods.***

**Method Signature:**

***public abstract void destroy();***

**4.getServletConfig():**

***=>getServletConfig()-method is used to display the servlet-configuration details.***

**Method Signature:**

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

**5.getServletInfo():**

***=>getServletInfo()-method will servlet information.***

**Method Signature:**

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

**Note:**

***=>The following three methods are known as Servlet Life-Cycle methods,because they are  
executed automatically in the same order:***

***(i)init()***

(ii)service()

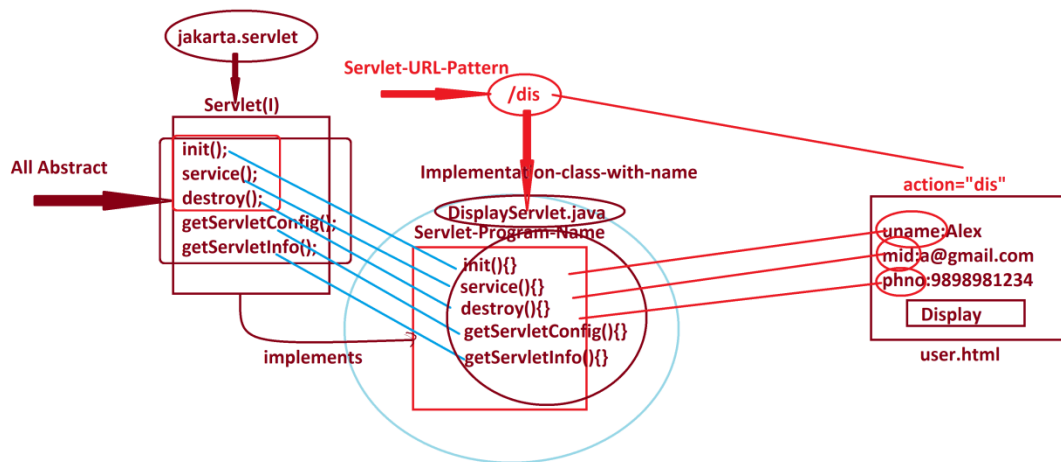
(iii)destroy()

**Note:**

=>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.

(Which means Construct bodies for all abstract methods of Servlet-Interface)

**Diagram:**



**\*imp**

**Construct Servlet Application using IDE Eclipse:**

**step-1 : Open IDE Eclipse,while opening name 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' to Project through 'Build path'**

**RightClick on Dynamic Web 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(one time process)**

**Click on Servers->click on 'Click this link to create a New Server'->select the server and**

**click 'Next'->Browse 'Tomcat Installation Directory'->select Folder->click 'Finish'**

**step-5 : Construct HTML file to read data to Servlet program**

**RightClick on webapp->New->HTML file->name the file->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 in 'WEB-INF'**

**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 class(Servlet program) to display user details.**

*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 ServletConfig getServletConfig()  
    {  
        return this.getServletConfig();  
    }  
    @Override  
    public String getServletInfo()  
    {  
        return "This Servlet will display User details";  
    }  
}
```

**step-9 : Execute the Application**

**RightClick on Application(Dynamic Web Project)->Run As->Run on Server->Select the Server->**

**Click 'Finish'**

[http://localhost:8082/App\\_Servlet\\_1/](http://localhost:8082/App_Servlet_1/)

=====

Venkatesh Maipathii