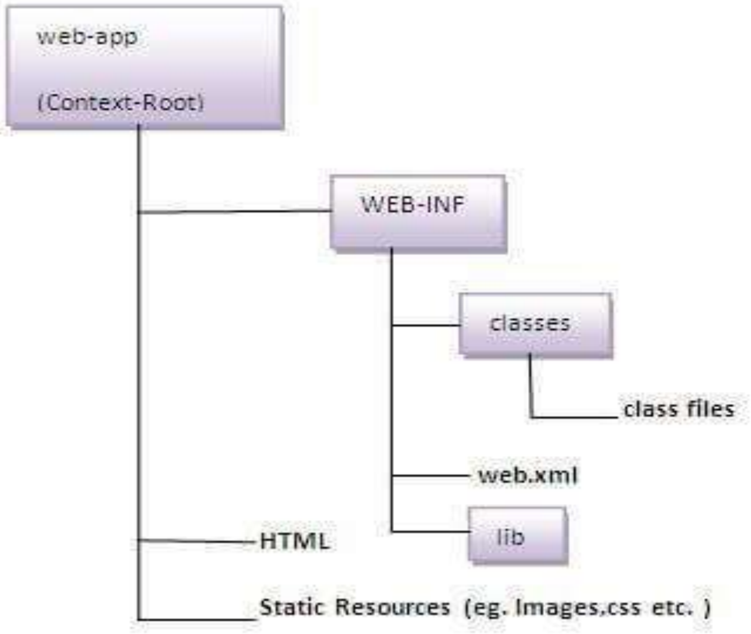
Servlet Implementation

Steps to create a servlet example

- There are given 6 steps to create a **servlet example**. These steps are required for all the servers.
  1. Create a directory structure
  2. Create a Servlet
  3. Compile the Servlet
  4. Create a deployment descriptor
  5. Start the server and deploy the project
  6. Access the servlet

1. Create a directory structure

- The **directory structure** defines that where to put the different types of files so that web container may get the information and respond to the client.



2. Create a Servlet

- There are three ways to create the servlet.
  - – By implementing the Servlet interface
  - – By inheriting the GenericServlet class
  - – By inheriting the HttpServlet class
- The HttpServlet class is widely used to create the servlet because it provides methods to handle http requests such as doGet(), doPost, doHead() etc.

```java
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;
public class DemoServlet extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{
res.setContentType("text/html");//setting the content type
PrintWriter pw=res.getWriter();//get the stream to write the data

//writing html in the stream
pw.println("<html><body>");
pw.println("Welcome to servlet");
pw.println("</body></html>");

pw.close();//closing the stream
}}
```

3. Compile the servlet

- For compiling the Servlet, jar file is required to be loaded. Different Servers provide different jar files:

Jar file	Server
1) servlet-api.jar	Apache Tomcat
2) weblogic.jar	Weblogic
3) javaee.jar	Glassfish
4) javaee.jar	JBoss

4. Create the deployment descriptor (web.xml file)

- The **deployment descriptor** is an xml file, from which Web Container gets the information about the servet to be invoked.

```xml
<web-app>

<servlet>
<servlet-name> Servlet 1</servlet-name>
<servlet-class>DemoServlet</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name> Servlet 1 </servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>

</web-app>
```

Description of the elements of web.xml file

<web-app> represents the whole application.

<servlet> is sub element of <web-app> and represents the servlet.

<servlet-name> is sub element of <servlet> represents the name of the servlet.

<servlet-class> is sub element of <servlet> represents the class of the servlet.

<servlet-mapping> is sub element of <web-app>. It is used to map the servlet.

<url-pattern> is sub element of <servlet-mapping>. This pattern is used at client side to invoke the servlet.

5. Start the Server and deploy the project

There are several ways to deploy the project. They are as follows:

- By copying the context(project) folder into the webapps directory
- By copying the war folder into the webapps directory
- By selecting the folder path from the server
- By selecting the war file from the server

6. How to access the servlet

- Open broser and write

  http://hostname:portno/contextroot/urlpatternofservlet.

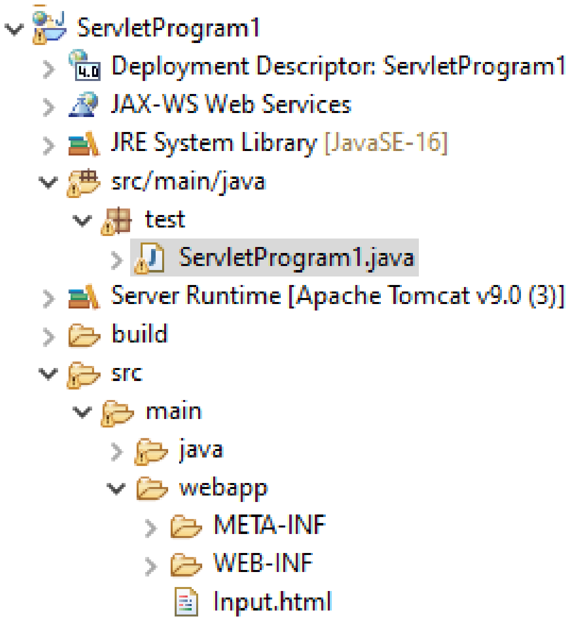
- For example:

  http://localhost:9999/demo/welcome

Example1

Input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method="post" action="pqr">
Name: <input type="text" name="name">
<input type="submit" value="Display">
</form>

</body>
</html>
```



```
ServletProgram1
  Deployment Descriptor: ServletProgram1
  JAX-WS Web Services
  JRE System Library [JavaSE-16]
  src/main/java
    test
      ServletProgram1.java
  Server Runtime [Apache Tomcat v9.0 (3)]
  build
  src
    main
      java
      webapp
        META-INF
        WEB-INF
        Input.html
```

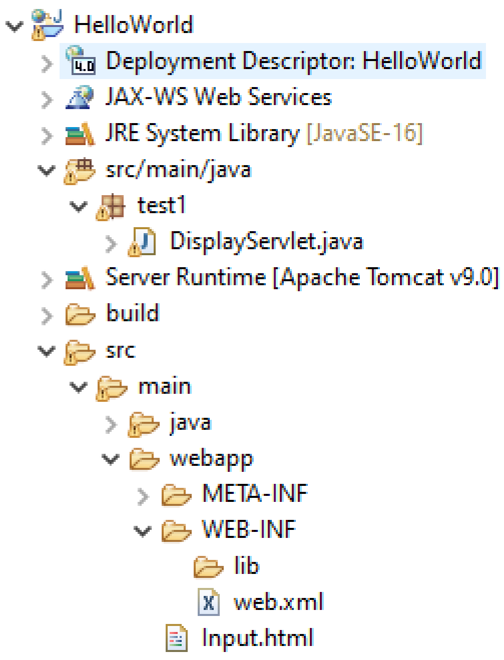
ServletProgram1.java

```
package test;
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
@WebServlet("/pqr")
public class ServletProgram1 extends HttpServlet{
   public void init() { }
   public void service(ServletRequest req, ServletResponse res)throws
   IOException, ServletException {
            PrintWriter pw=res.getWriter();
            res.setContentType("text/html");
            pw.println(".....WELCOME*****");
   }
   public void destroy() {

   }
}
```

Example2

Input.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method="post" action="xyz">
Name: <input type="text" name="name">
<input type="submit" value="Display">
</form>

</body>
</html>
```



```
HelloWorld
  Deployment Descriptor: HelloWorld
  JAX-WS Web Services
  JRE System Library [JavaSE-16]
  src/main/java
    test1
      DisplayServlet.java
  Server Runtime [Apache Tomcat v9.0]
  build
  src
    main
      java
      webapp
        META-INF
        WEB-INF
          lib
          web.xml
        Input.html
```

DisplayServlet.java:

```
package test1;
import java.io.*;
import java.io.PrintWriter;

import javax.servlet.*;
import javax.servlet.http.*;
public class DisplayServlet  extends HttpServlet{
    public void doPost(HttpServletRequest req, HttpServletResponse
    res)throws ServletException, IOException{
            PrintWriter pw=res.getWriter();
            res.setContentType("text/html");
            String name=req.getParameter("name");
            pw.println("name is  "+name);
    }

}
```

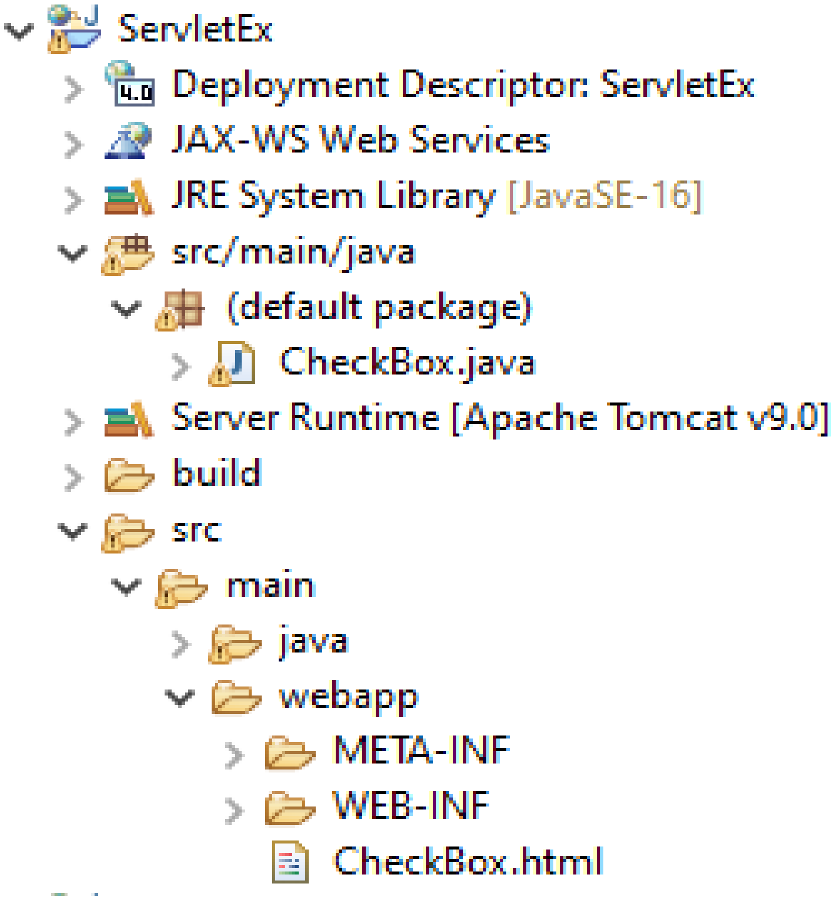
web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="http://xmlns.jcp.org/xml/ns/javaee"
   xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
   http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd" id="WebApp_ID"
   version="4.0">
 <display-name>HelloWorld</display-name>
 <welcome-file-list>
            <welcome-file>Input.html</welcome-file>
  </welcome-file-list>
 <servlet>
            <servlet-name>ds</servlet-name>
            <servlet-class>test1.DisplayServlet</servlet-class>
 </servlet>
 <servlet-mapping>
            <servlet-name>ds</servlet-name>
            <url-pattern>/xyz</url-pattern>
 </servlet-mapping>

 </web-app>
```

Example3

CheckBox.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action = "CheckBox" method = "POST" target = "_blank">
        <input type = "checkbox" name = "maths" checked = "checked" /> Maths
        <input type = "checkbox" name = "physics"  /> Physics
        <input type = "checkbox" name = "chemistry" checked = "checked" />
                                        Chemistry
        <input type = "submit" value = "Select Subject" />
      </form>
</body>
</html>
```



- ServletEx
  - Deployment Descriptor: ServletEx
  - JAX-WS Web Services
  - JRE System Library [JavaSE-16]
  - src/main/java
    - (default package)
      - CheckBox.java
  - Server Runtime [Apache Tomcat v9.0]
  - build
  - src
    - main
      - java
      - webapp
        - META-INF
        - WEB-INF
        - CheckBox.html

CheckBox.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
@WebServlet("/CheckBox")
// Extend HttpServlet class
public class CheckBox extends HttpServlet {

  // Method to handle GET method request.
  public void doGet(HttpServletRequest
     request, HttpServletResponse response)
    throws ServletException, IOException {

    // Set response content type
    response.setContentType("text/html");

    PrintWriter out = response.getWriter();
    String title = "Reading Checkbox Data";
    String docType =
      "<!DOCTYPE html>";
    out.println(docType +
      "<html>\n" +
        "<head><title>" + title + "</title></head>\n" +
        "<body bgcolor = \"#f0f0f0\">\n" +
          "<h1 align = \"center\">" + title + "</h1>\n" +
          "<ul>\n" +
            "  <li><b>Maths Flag : </b>: "
            + request.getParameter("maths") + "\n" +
            "  <li><b>Physics Flag: </b>: "
            + request.getParameter("physics") + "\n" +
            "  <li><b>Chemistry Flag: </b>: "
            + request.getParameter("chemistry") + "\n" +
          "</ul>\n" +
        "</body>"+"</html>");
  }

  // Method to handle POST method request.
  public void doPost(HttpServletRequest request,
     HttpServletResponse response)
    throws ServletException, IOException {

    doGet(request, response);
  }
}
```