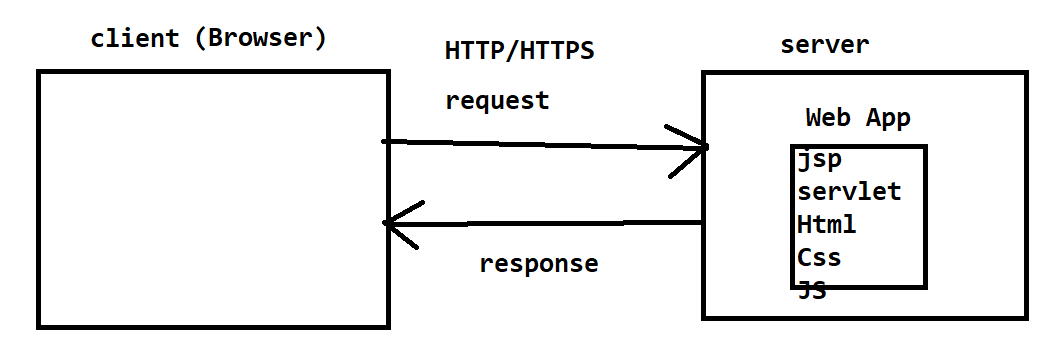
**Servlet and JSP**

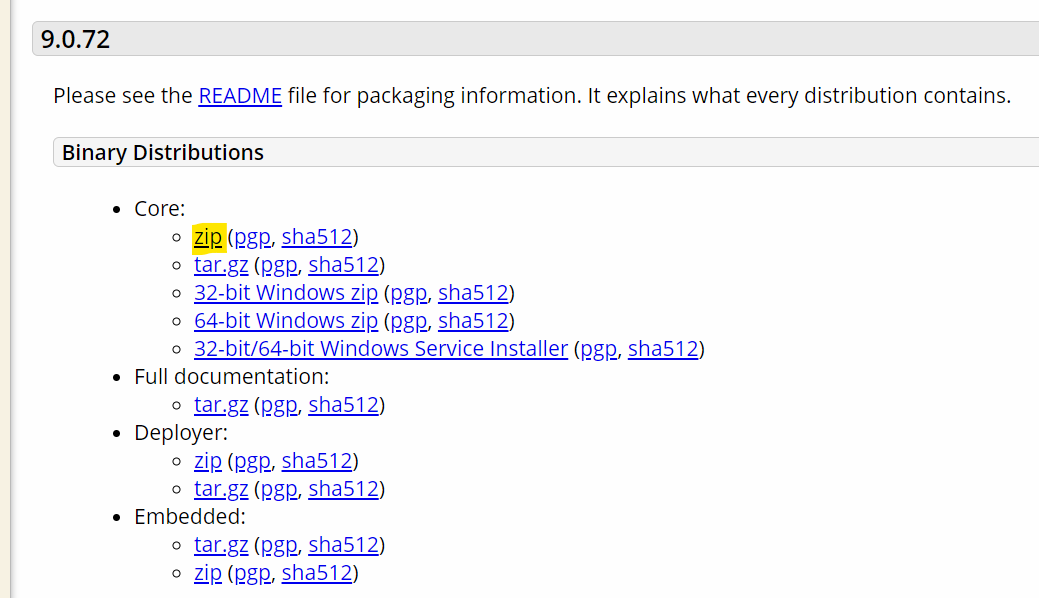
1. Using JSP servlet you can develop the web applications.
2. These applications are also known as client-server application
3. To Run type of application you will required the server along with JDK
4. In these application you will use a client and server side technologies.
5. Jsp and servlet are the server side technologies.



**Server Download and Setup**

1. Download Tomcat Server

<https://tomcat.apache.org/download-90.cgi>



1. Server Setup
   1. Extract a ZIP file into appropriate location.
   2. Setup The Server into Eclipse
      1. Open an eclipse
      2. Set Eclipse perspective to ‘Java EE’
      3. Go To ‘Servers’ tab at the bottom of the screen
      4. Click on Create new Server Option
      5. Expand the ‘Apache’ option
      6. Select the Apache version form the list
      7. Click on Next.
      8. Set the installation directory
         1. Browse a location where you extracted you zip file.
         2. Select the parent folder of bin, config, lib etc.
      9. Click on Next and then Finish

**Web Application**

* + - 1. The application which can be execute from the browser over an network.
      2. Web Applications are of 2 types
         1. Static Web Application

The content of the page is common for all the users

These application are informative web sites.

* + - * 1. **Dynamic Web application**

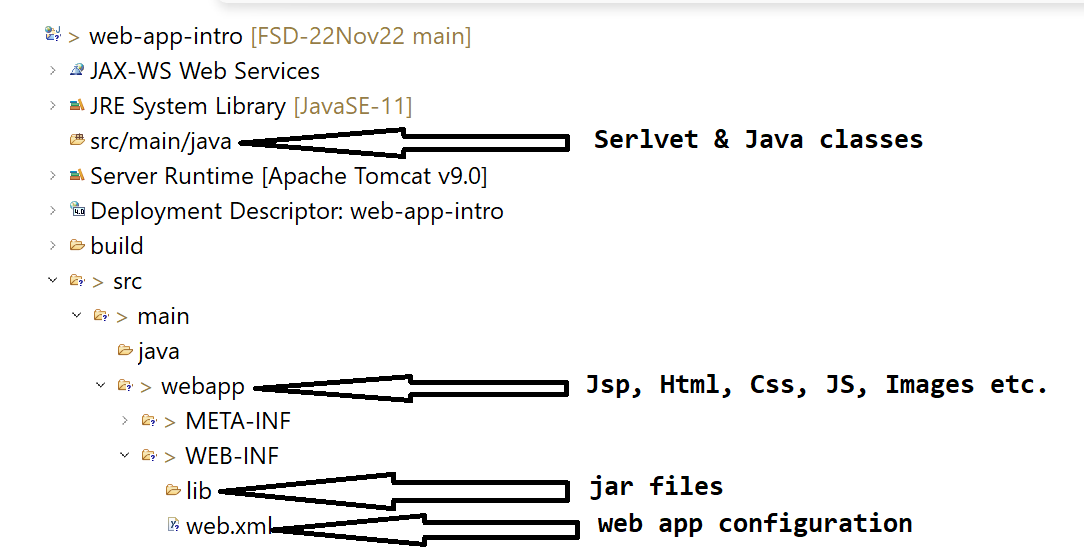
The content of the page is different for every user.

In this application you can perform the activities which is specific to the user.

**Create and Execute Dynamic Web Project**

1. Go To ‘New’ -> click on ‘Dynamic Web Project’
2. Set the name of the project.
3. **Make Sure than Target Runtime is not <NONE> and is must be selected (Tomcat server)**
4. Click on Next and Next
5. Select the check box for Deployment Descriptor.
6. Click on Finish

**Dynamic Web Project Structure**

****

**Protocol**

TCP, IP, FTP, FTPS, SMTP, UDP, **HTTP, HTTPS**

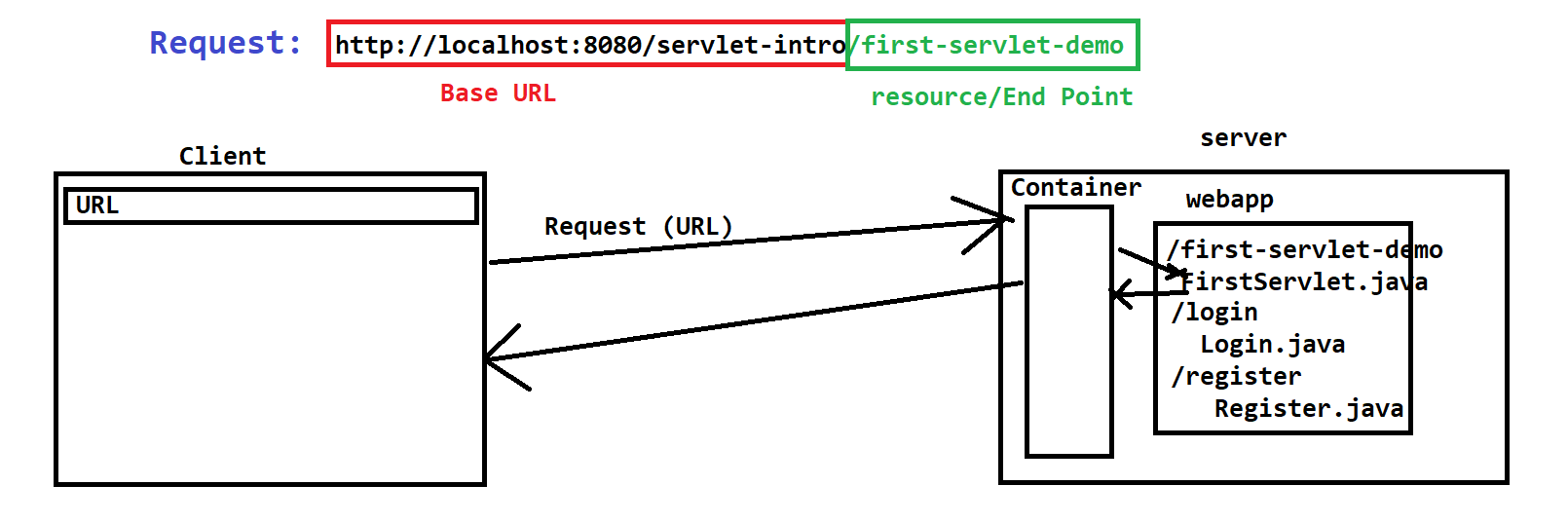
**Servlet**

1. Servlets are the java classes.
2. Servlet do not have main methods.
3. Main Purpose of using this is to get the request, process the request and generate the response.
4. Servlet is use to create dynamic web pages.
5. Inside servlet you can use the Java code as well as HTML code.
6. The HTML code has to add inside the java code and hence it is also known as HTML in JAVA
7. Every servlet will be access using an URL. There must be a unique URL provided to a servlet.
8. Servlet is maintain by the server, that is object creation and object management will be taken case by Server.
9. There is only one object of the servlet created in an application.

**How to create Servlet**

1. There 3 options to create servlet
   1. By extending **Servlet** class
   2. By Extending **GenericServlet** class
   3. By Extending **HttpServlet** class
2. Create a java class use any one of the class as a super class.
3. Override the service method. And implement your request and response code inside the service method.
4. Provide the URL for the servlet so that client can access the servlet by using URL. To provide the URL you can use **@WebServlet** annotation on the class.

**Request and response processing flow**

****

**Returning the Response to client**

1. You will required the object if HttpServletResponse
2. You have to set the type of response which is also known as MIME type.

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types>

1. MIME type can be set using the response object method ‘**setContentType(“MIME\_TYPE”)**’
2. You have to get an object of PrintWriter (text type) or InputStream (binary) to write the response.

**PrintWriter out = response.getWriter();**

1. Write use response into the print() method.

**Request handling and Processing**

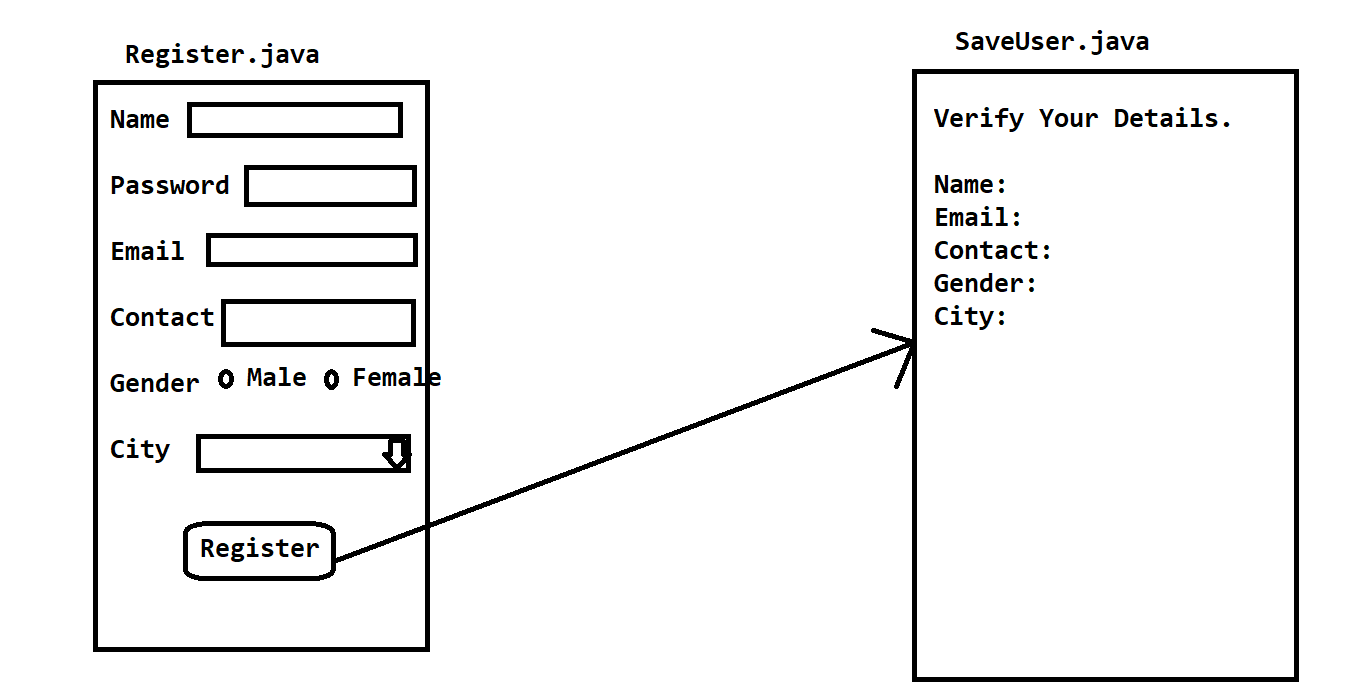
1. To Handle a request you can use the HttpServletRequest Object.
2. From the request Object you can get the data which is in the form of parameter or attribute.
3. Request can also be consider as a URL.

**Parameter**

1. Parameter is a user data.
2. Parameter pass from the request or from the request body.
3. Every parameter has a key and value.
4. Parameters are visible inside URL after ‘?’
5. These parameters can be access by request object.
6. Every Parameter is in the form of String only.
7. There can be a multiple parameters in the URL which must be separated by ‘&’
8. To Get the parameter from the request you can use a request object and getParameter method.
9. Syntax:

Request.getParameter(“Key”) : String

Task-1



**Add Jar file in web application**

1. Copy a jar file form the folder location.
2. Paste the jar file inside the application at location src/main/webapp/WEB-INF/lib