JDBC

**Externalization of Properties**

1. Create a resources folder inside project
   1. Right Click on Project -> “New” option -> “Source Folder” option
   2. Provide name “resources”
   3. Click on Finish
2. Create a properties file inside resources folder
   1. Right click on the resources folder
   2. “New” Option -> “File” option
   3. File Name with extension “.properties”
3. Create a properties inside the file
   1. Every property must has a key and value
   2. Key and value can be separated by “=” or “:”
   3. Every property must be on new line.

**Java Editions**

JSE

Java Standard Edition (JSE) also known as Core Java

Can develop Console based and Desktop application.

Example: Eclipse, NetBeans, Oracle

JEE

Java Enterprise Edition (JEE) also known as Advance Java

JSP, Servlet, JMS, JSR, WebServices etc.

Can Develop Web Application.

Example: IRCTC, SBI, ICICI etc..

JME

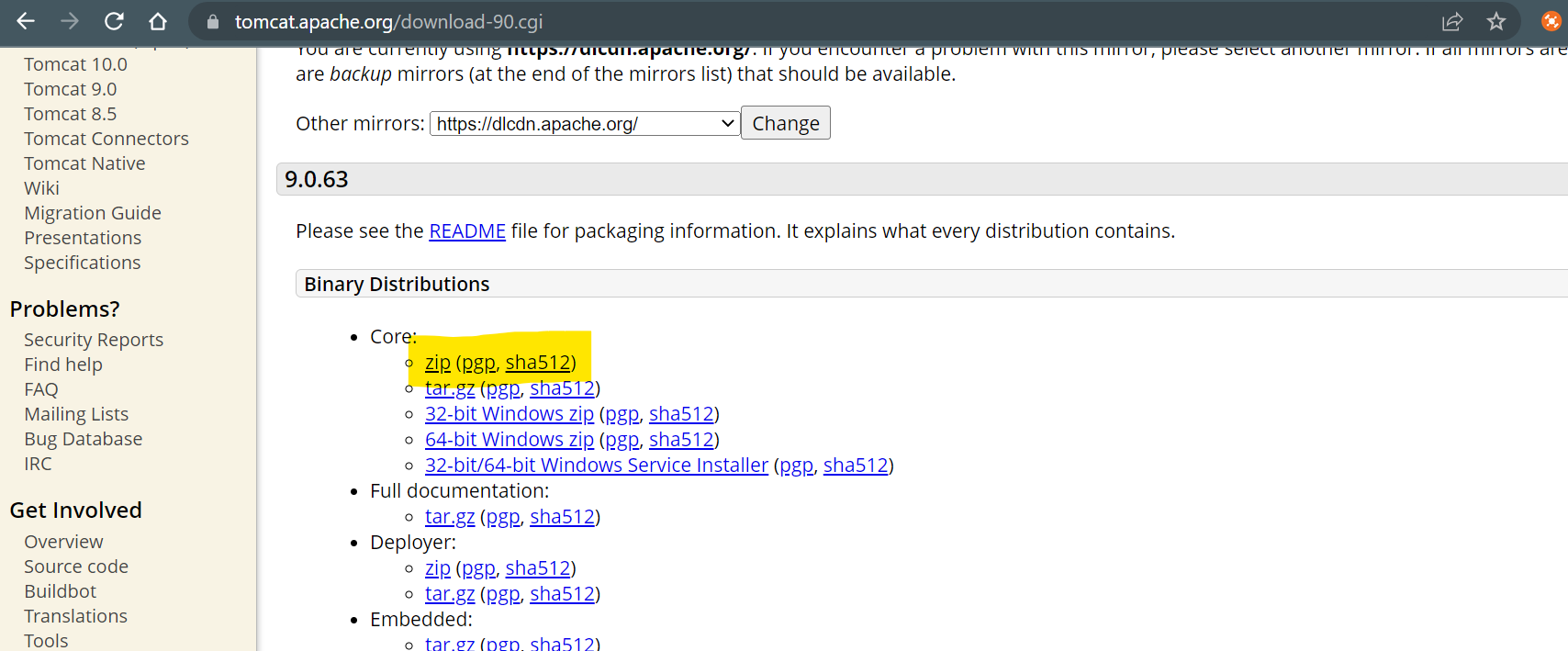
Java Micro Edition (JME).

Use to develop Embedded application, Mobile Application.

**Server Setup**

1. Download Server
   1. Tomcat Server Download

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



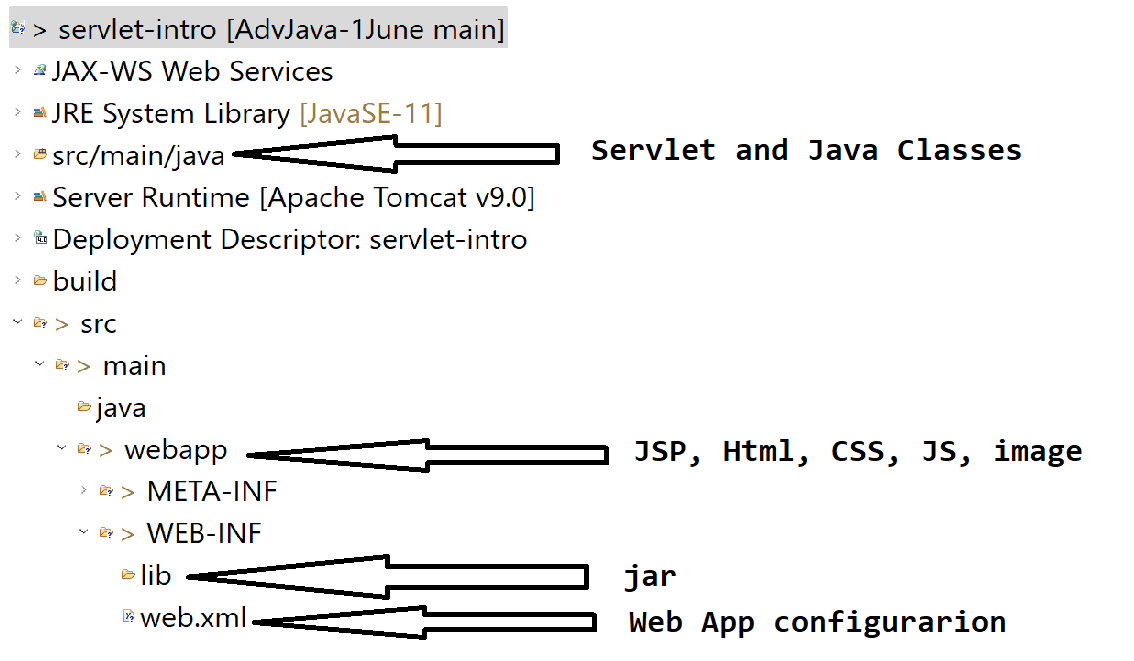
* 1. Extract the ZIP file
  2. To Start server Go to “bin” -> double click on “startup.bat”

1. Configure Server into Eclipse
   1. Open eclipse workspace
   2. Set a Java EE perspective.
   3. Add Server into Servers tab at the bottom of the window
   4. Click On the “Create New Server” link
   5. Expand The “Apache” Option from the new window and Select The downloaded tomcat version
   6. Click on “Next”
   7. Browser the director of the tomcat folder (Select a parent folder of the bin, lib, config etc.)
   8. “Next” and “Finish”
2. Start The Tomcat Server
   1. Right click on Server and click on the start.

**Dynamic Web Application**

1. File -> New –> “Dynamic Web Project”
2. Set a Project name and Make sure than Target runtime must not be “<None>”
3. Click Next - > Next
4. Make sure that select check box for “web.ml deployment descriptor”

**Dynamic Project Structure**



**JSP and Servlet**

1. Used to develop a dynamic web application
2. These technologies are executes at server side.
3. In JSP and Servlet you can get request and generate response

**Servlet**

1. Servlet is a java class.
2. Used to create dynamic Web Page
3. Servlet do not have main method.
4. Servlet are executes by the Servlet container which is a part of server.
5. Servlet can read a request, process a request and generate response.
6. Every Servlet must have a unique URL and it has to register.
7. Servlet Object creation and maintenance will be done by Servlet Container.
8. There is only one object of servlet gets created in a servlet container.
9. Servlet can contains CSS and HTML code along java code.
10. Servlet is also known as HTML in Java.

**How To Create Servlet**

1. There are 3 options to create servlet
   1. By Implementing **Servlet** interface
   2. Extending **GenericServlet** abstract class
   3. Extending **HttpServlet** abstract class
2. Can Override a service method to process request and generate response.

**Register Servlet**

1. By Using XML
   1. This configuration has to do inside web.xml (src/main/webapp/WEB-INF)
   2. XML Tags to configure Servlet

<servlet>

<servlet-name>first</servlet-name>

<servlet-class>FirstServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>first</servlet-name>

<url-pattern>/first-servlet</url-pattern>

</servlet-mapping>

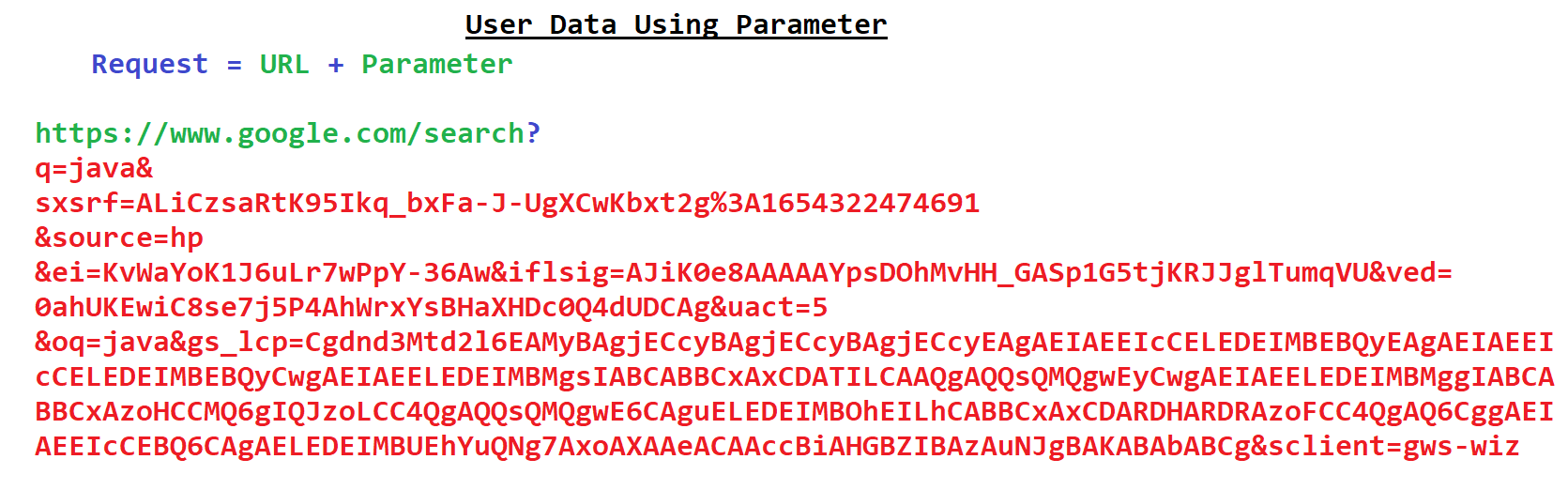
1. Using Annotation
   1. Annotation Configuration can be achieved by using @WebServlet Annotation on the servlet class.
   2. Syntax:

@WebServlet("/<URL>")

**Request and Response**

**Request**

1. Request always goes from Client to Server
2. Request can have a user data.
3. This Data can be pass in form of parameter or request body.
4. Parameter :



**Response**

1. Flows from server to Client
2. Types of response also known as MIME type

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

**Servlet Life Cycle**

1. Manage by Servlet container.
2. There are three stages of the servlet life cycle
   1. **Init stage**
      1. This stage executes when container creates an object of the servlet.
      2. As a part of this stage **init(ServletConfig)** gets invoked.
      3. Every servlet gets instantiated (Object) only once in a life cycle and hence this methos gets called only once.
      4. In this stage you can perform the initialization activity.
      5. This method gets called after constructor.
   2. **Service Stage**
      1. This stage gets executed for every request of the user.
      2. This stage gets executes multiple time in life cycle.
      3. As a part of this stage **service(HttpServletRequest, HttpServletResponse)** method gets invoked.
      4. This stage is use to accept the request, process the request data and generate the response.
   3. **Destroy Stage** 
      1. This stage gets executed only once in a life cycle.
      2. As a part of this stage **destroy()** method gets invoked.
      3. In this stage all the cleanup activities can be perform.