Java EE - Day 1

From browser application:

http://localhost:8081/demo-web-app/

Received by Server running locally and listening at port: 8081 ie. Tomcat

Now Tomcat verifies whether demo-web-app application is deployed ? YES

Next Step, any pattern after demo-web-app/ ? NO

Next Step: Go to web.xml, check welcome file list, if any the file names
Specified in web.xml, is it present under web app folder ? YES

Now Tomcat will serve that file to the client/browser

Note: Project name is also called as context root

Static Pages or Dynamic Pages ?

home.html and default.html are static pages

How to create dynamic pages:

We require Scripting languages to generate content dynamically:

1. Java. 2. Python.

—----------

The two widely used technologies for developing dynamic web pages are:

1.Servlets
2.JSP –Java Server Pages

***A* Servlet** is a specialized Java class that runs on a web server to generate dynamic web pages.
Servlets work on a request -response programming model i.e. they accept requests from the clients , generate responses dynamically and send the responses in a format such as ***html*** to the clients/browsers.

```
package com.wipro.controller;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/*
* http://localhost:8081/dynamic-web-app-demo/hello
*
* Since default HTTP method is GET, control enters into gdoGet() method
*
*    HttpServlet -> GenericServlet -----> Servlet
*
*    Servlet is an interface, GenericServlet is a class that implements
Servlet,
*    HttpServlet is class that extends GenericServlet.
*
*
```

```
*    GenericServlet can handle any type of protocol( Http, ftp, SMTP etc) whereas HttpServlet
*    can handle only Http protocol. Since most of the web applications are http-base, we create our own
*    servlet that extend HttpServlet.
*
*/
@WebServlet("/hello")
public class HelloWorldServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		PrintWriter out = response.getWriter();
		//dynamically generate the web page, writes into response object
		out.println("<html><body><h1><font color='red'> Welcome to My Dynamic Page</font></h1></body></html>");
	}

	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		doGet(request, response);
	}
}


——----------------
<servlet>
  <description></description>
  <display-name>HelloWorldServlet</display-name>
  <servlet-name>HelloWorldServlet</servlet-name>
  <servlet-class>com.wipro.controller.HelloWorldServlet</servlet-class>
```

```
 </servlet>
 <servlet-mapping>
   <servlet-name>HelloWorldServlet</servlet-name>
   <url-pattern>/hello</url-pattern>
 </servlet-mapping>
```

—----------------------------------

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
                <h2>Login Form</h2>
                <form action="" method="get">
                        Enter Userid: <input   type="text" name="userid"
size="20" />
                        Enter Password: <input type="text" name="password"
size="20"/>
                        <input type="submit"  value="Login"/>
                </form>
</body>
</html>
```

—----------------------------------

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
```

```
            <h2>Login Form</h2>
            <form action="login" method="post">
                Enter Userid: <input   type="text" name="userid"
size="20" /> <br>
                Enter Password: <input type="password"
name="password"  size="20"/><br>
                <input type="submit"  value="Login"/>
            </form>
</body>
</html>
```

```
package com.wipro.controller;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/*
* If Http method is GET, the form data is appended to the url and
sent to the server in the foll. format:
*
*                   url?querystring
*
*          query string format:
*                   name=value&name=value
*
*      Ex.
```

```
*
http://localhost:8081/dynamic-web-app-demo/login?userid=Srini&password=Srini%40123
*
*		While sending sensitive data or large amount of data to the server, apply HttP POST/PUT methods
*/
@WebServlet("/login")
public class LoginServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;


	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		try {
			PrintWriter out = response.getWriter();
			String userid = request.getParameter("userid");
			String password = request.getParameter("password");

			System.out.println(userid+","+password);

			out.println("<html><body><h2>Your Credentials:"+ userid+","+password+"</h2></body></html>");


		}catch(Exception e) {
```

```
response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR,e.getMessage());
                }

        }

        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

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

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<!-- Font Awesome -->
<link

href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css"
 rel="stylesheet"
/>
<!-- Google Fonts -->
<link
```

```
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap"
 rel="stylesheet"
/>
<!-- MDB -->
<link

href="https://cdnjs.cloudflare.com/ajax/libs/mdb-ui-kit/7.3.0/mdb.min.css"
 rel="stylesheet"
/>
<style>
        .intro {
 height: 100%;
}
@media (min-height: 300px) and (max-height: 450px) {
 .intro {
   height: auto;
 }
}
.gradient-custom {
 background: radial-gradient(50% 123.47% at 50% 50%, #00FF94 0%, #720059 100%), linear-gradient(121.28deg, #669600 0%, #FF0000 100%), linear-gradient(360deg, #0029FF 0%, #8FFF00 100%), radial-gradient(100% 164.72% at 100% 100%, #6100FF 0%, #00FF57 100%), radial-gradient(100% 148.07% at 0% 0%, #FFF500 0%, #51D500 100%);
 background-blend-mode: screen, color-dodge, overlay, difference, normal;
```

```
}
</style>
</head>
<body>
<section class="intro">
 <div class="mask d-flex align-items-center h-100"
style="background-color: #D6D6D6;">
   <div class="container">
    <div class="row justify-content-center">
      <div class="col-12 col-md-8 col-lg-6 col-xl-5">
       <div class="card" style="border-radius: 1rem;">
        <div class="card-body p-5 text-center">
         <div class="my-md-5 pb-5">
          <h1 class="fw-bold mb-0">Welcome</h1>
          <i class="fas fa-user-astronaut fa-3x my-5"></i>
                        <form action="login" method="post">
                  <div class="form-outline mb-4">
                   <input type="email" id="typeEmail" name="email"
class="form-control form-control-lg" />
                   <label class="form-label"
for="typeEmail">Email</label>
                  </div>

                  <div class="form-outline mb-5">
                   <input type="password" id="typePassword"
name="password" class="form-control form-control-lg" />
                   <label class="form-label"
for="typePassword">Password</label>
                  </div>
```

```
                    <button class="btn btn-primary btn-lg btn-rounded
gradient-custom text-body px-5" type="submit">Login</button>

                </div>
                        </form>
            <div>
              <p class="mb-0">Don't have an account? <a href="#!"
class="text-body fw-bold">Sign Up</a></p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
 </div>
</section>
</body>
</html>
```

```
package com.wipro.controller;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/*
* If Http method is GET, the form data is appended to the url and
sent to the server in the foll. format:
```

```
*
*                       url?querystring
*
*           query string format:
*                       name=value&name=value
*
*       Ex.
*
http://localhost:8081/dynamic-web-app-demo/login?userid=Srini&
password=Srini%40123
*
*           While sending sensitive data or large amount of data to
the server, apply HttP POST/PUT methods
*/
@WebServlet("/login")
public class LoginServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;


        protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {
                try {
                        PrintWriter out = response.getWriter();
                        String email = request.getParameter("email");
                        String password =
request.getParameter("password");

                        System.out.println(email + "," +password);
```

```
                    out.println("<html><body><h2>Your Credentials:"+
email+","+password+"</h2></body></html>");


                }catch(Exception e) {

response.sendError(HttpServletResponse.SC_INTERNAL_SERVER_ERROR,e.getMessage());
                }

        }

        protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {

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

—----------

ServletConfig and ServletContext objects.

For every Servlet object, Servlet Container implicitly create one ServletConfig object which contains initialization values provided to the Servlet..

When a first request for a Servlet comes from the client, Servlet container instantiates the Servlet class, instance/object of the servlet is created and

at that time if we provide some initialization values, those values values will be placed in ServletConfig object.

When a second request comes for the same Servlet , will another instance be created? NO.

A new thread is spawned and that thread will enter into the servlet and execute doGet()/doPost() methods.

For all subsequent threads, separate threads are spawned and they execute the method(s) of the Servet.

Servlet Lifecycle methods:

init() : Executed only once during instantiation of the servlet.
service() : converted into doGet()/doPost() .. methods in HttpServlet class get executed for every request/thread.
destroy() : executed only once when the servlet is removed from servlet container

```
package com.wipro.controller;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletConfig;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebInitParam;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/*
* http://localhost:8081/dynamic-web-app-demo/hello
*
```

```
* Since default HTTP method is GET, control enters into gdoGet()
method
*
*    HttpServlet -> GenericServlet -----> Servlet
*
*    Servlet is an interface, GenericServlet is a class that
implements Servlet,
*    HttpServlet is class that extends GenericServlet.
*
*
*    GenericServlet can handle any type of protocol( Http, ftp,
SMTP etc) whereas HttpServlet
*    can handle only Http protocol. Since most of the web
applications are http-base, we create our own
*    servlet that extend HttpServlet.
*
*/
@WebServlet(urlPatterns= "/hello" ,
                    initParams = {
                                        @WebInitParam(name =
"user", value = "Srini"),
                                        @WebInitParam(name =
"job", value = "Trainer")
                                    }
                    )
public class HelloWorldServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;
```

```
	protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {

		PrintWriter out = response.getWriter();
		//dynamically generate the web page, writes into
response object
		out.println("<html><body><h1><font color='red'>
Welcome to My Dynamic Page</font></h1></body></html>");

		//getting reference to ServletConfig object of
HelloWorldServlet instance
		ServletConfig config = this.getServletConfig();
		//ServletConfig object contains initialization values of
the servlet
		out.println(config.getInitParameter("user"));
		out.println(config.getInitParameter("job"));


	}

	protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {

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

ServletContext object:

There is only one ServletContext object per web application.

The data stored in ServletContext object is global in the sense all the web components the application can access the contents of ServletContext object.

SerlvetContext is created implicitly when the web application is deployed on to the web server and implicitly removed when the web application is undeployed from the container.

How to initialize a ServletContext object? **We can initialize only in web.xml file**

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd"
id="WebApp_ID" version="6.0">
 <display-name>dynamic-web-app-demo</display-name>
 <welcome-file-list>
   <welcome-file>login.html</welcome-file>
   <welcome-file>index.jsp</welcome-file>
   <welcome-file>index.htm</welcome-file>
   <welcome-file>default.html</welcome-file>
   <welcome-file>default.jsp</welcome-file>
```

```
  <welcome-file>default.htm</welcome-file>
</welcome-file-list>

<context-param>
      <param-name>database</param-name>
      <param-value>Oracle</param-value>
</context-param>
</web-app>
```

```
ServletContext context = this.getServletContext();

out.println("<br><h2>"+context.getInitParameter("database")+"</h2>");
```

The Servlet API objects covered so far are:

Explicitly created:

1. HttpServlet object : Custom servlets

Implicitly created

2. HttpServletRequest object : contains information coming from client
   application(browser, postman, Swagger etc)
3. HttpServletResponse object: contains information that is sent to client
   from the Server.
4. ServletConfig object
5. ServletContext object

Apart from initialization, we can also **explicitly store data** in the following objects:

1. HttpServletRequest object
2. HttpServletSession object
3. ServletContext object

The following methods:

getAttribute()
setAttribute()
removeAttribute()

```
package com.wipro.controller;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletContext;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
@WebServlet("/attr")
public class AttributeServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;
```

```
	protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {

		PrintWriter out = response.getWriter();
		request.setAttribute("greeting", "Hi, Welcome to my
home");
		request.setAttribute("dinner", "Great meal tonight");

		ServletContext context= this.getServletContext();
		context.setAttribute("college","MGM College of
Engineering");



		out.println("<br>"+ request.getAttribute("greeting"));
		out.println("<br>"+ request.getAttribute("dinner"));



		out.println("<br><br>"+ context.getAttribute("college"));

	}

	protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {

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

—----------

```
package com.wipro.controller;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletConfig;
import jakarta.servlet.ServletContext;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebInitParam;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/*
* http://localhost:8081/dynamic-web-app-demo/hello
*
* Since default HTTP method is GET, control enters into gdoGet() method
*
*    HttpServlet -> GenericServlet -----> Servlet
*
*    Servlet is an interface, GenericServlet is a class that implements Servlet,
*    HttpServlet is class that extends GenericServlet.
*
*
*    GenericServlet can handle any type of protocol( Http, ftp, SMTP etc) whereas HttpServlet
*    can handle only Http protocol. Since most of the web applications are http-base, we create our own
```

```
*    servlet that extend HttpServlet.
*
*/
@WebServlet(urlPatterns= "/hello" ,
                    initParams = {
                                        @WebInitParam(name =
"user", value = "Srini"),
                                        @WebInitParam(name =
"job", value = "Trainer")
                                    }
                    )
public class HelloWorldServlet extends HttpServlet {
      private static final long serialVersionUID = 1L;

      protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {

            PrintWriter out = response.getWriter();
            //dynamically generate the web page, writes into
response object
            out.println("<html><body><h1><font color='red'>
Welcome to My Dynamic Page</font></h1></body></html>");

            //getting reference to ServletConfig object of
HelloWorldServlet instance
            ServletConfig config = this.getServletConfig();
            //ServletConfig object contains initialization values of
the servlet
            out.println(config.getInitParameter("user"));
```

```
            out.println(config.getInitParameter("job"));

            ServletContext context = this.getServletContext();

out.println("<br><h2>"+context.getInitParameter("database")+"</h
2>");


out.println("<br><h2>"+context.getAttribute("college")+"</h2>");



        }

        protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException {

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

—----------