Difference between GET and POST

Step 1: Creating a dynamic web project

- Open Eclipse
- Go the File menu. Choose New->Dynamic Web Project
- Enter the project name as ServletGetPost. Click on Next
- Enter nothing in the next screen and click on Next
- Check the checkbox Generate web.xml deployment descriptor and click on Finish
- This will create the project files in the Project Explorer

Step 2: Creating an HTML page

- In the Project Explorer, expand the project ServletGetPost
- Expand WebContent. Right click on WebContent. Choose New-> HTML File
- Enter the filename as index.html and click on Finish
- Enter the following code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>GET vs POST Demo</title>
</head>
<body>

<a href="gethandler?name=John+Doe&address=145+Lake+Avenue">Do a
Servlet GET</a><br><br>
<form name=frm method=post action="posthandler">
  Name <input name="name" id="name" maxlenght=50><br>
  Address <input name="address" id="address" maxlength=100><br>
  <button>Submit</button>
</form>

</body>
</html>
```

- Click on the Save icon

Step 3: Creating a servlet GetHandler.java

- In the Project Explorer, expand ServletGetPost->Java Resources
- Right click src and choose New->Servlet
- In Class Name, enter GetHandler and click on Finish
- Enter the following code:

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

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

/**
* Servlet implementation class GetHandler
*/
@WebServlet("/GetHandler")
public class GetHandler extends HttpServlet {
	private static final long serialVersionUID = 1L;

   /**
* @see HttpServlet#HttpServlet()
*/
   public GetHandler() {
      super();
      // TODO Auto-generated constructor stub
   }

      /**
       * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
       */
      protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
            // TODO Auto-generated method stub
              PrintWriter out = response.getWriter();
               out.println("<html><body>");

               String name = request.getParameter("name");
               String address = request.getParameter("address");
               out.println("Name=" + name + "<br>Address=" + address);
               out.println("</body></html>");
      }

      /**
       * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
       */
      protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
            // TODO Auto-generated method stub
            doGet(request, response);
      }

}
```

Step 4: Creating a servlet PostHandler.java

- In the Project Explorer, expand ServletGetPost->Java Resources
- Right click src and choose New->Servlet
- In Class Name, enter PostHandler and click on Finish
- Enter the following code:

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

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
* Servlet implementation class PostHandler
*/
@WebServlet("/PostHandler")
public class PostHandler extends HttpServlet {
    private static final long serialVersionUID = 1L;

  /**
* @see HttpServlet#HttpServlet()
*/
  public PostHandler() {
    super();
    // TODO Auto-generated constructor stub
  }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
                // TODO Auto-generated method stub

    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                  PrintWriter out = response.getWriter();
                    out.println("<html><body>");

                    String name = request.getParameter("name");
                    String address = request.getParameter("address");

                    out.println("Name=" + name + "<br>Address=" + address);
                    out.println("</body></html>");
    }

}
```

Step 5: Configuring web.xml

- In the Project Explorer, expand ServletGetPost->WebContent->WEB-INF
- Double click web.xml to open it in the editor

- Enter the following script:

```
<?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>LoginLogout</display-name>
 <welcome-file-list>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>default.html</welcome-file>
  <welcome-file>default.htm</welcome-file>
  <welcome-file>default.jsp</welcome-file>
 </welcome-file-list>
 <servlet>
  <servlet-name>GetHandler</servlet-name>
  <servlet-class>GetHandler</servlet-class>
 </servlet>
 <servlet>
  <servlet-name>PostHandler</servlet-name>
  <servlet-class>PostHandler</servlet-class>
 </servlet>
 <servlet-mapping>
  <servlet-name>GetHandler</servlet-name>
  <url-pattern>/gethandler</url-pattern>
 </servlet-mapping>
 <servlet-mapping>
  <servlet-name>PostHandler</servlet-name>
  <url-pattern>/posthandler</url-pattern>
 </servlet-mapping>
</web-app>
```

Step 6: Checking for servlet-api.jar

- Before building the project, we need to add servlet-api.jar to the project
- Servlet-api.jar file is already present in your practice lab. (Refer FSD: Lab Guide - Phase 2)
- To add it to the project, follow the below mentioned steps:
  - In the Project Explorer, right click on ServletGetPost and choose Properties
  - Select Java Build Path from the options on the left
  - Click on Libraries tab on the right
  - Under ClassPath, expand the node that says Apache Tomcat
  - If there is an existing entry for servlet-api.jar, then click on Cancel and exit the window
  - If it is not there, then click on Classpath entry and click on Add External JARs button on the right
  - From the file list, select servlet-api.jar file and click Ok
  - Click on Apply and Close

Step 7: Building the project

- From the Project menu at the top, click on Build
- If any compile errors are shown, fix them as required

Step 8: Publishing and starting the project

- If you do not see the Servers tab near the bottom of the IDE, go to the Window menu and click Show View->Servers

- Right click on the Server entry and choose Add and Remove
- Click the Add button to move ServletGetPost from the Available list to the Configured List
- Click Finish
- Right click on the Server entry and click on Publish
- Right click on the Server entry and click on Start
- This will start the server

Step 9: Running the project

- To run the project, open a web browser and type: http://localhost:8080/ServletGetPost

Step 10: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

```
cd <folder path>
```

- Initialize your repository using the following command:

```
git init
```

- Add all the files to your git repository using the following command:

  ```
  git add .
  ```

- Commit the changes using the following command:

  ```
  git commit .  -m “Changes have been committed.”
  ```

- Push the files to the folder you initially created using the following command:

  ```
  git push -u origin master
  ```