

Setup of Eclipse with new Jakarta EE support on latest Tomcat: and creation of Dynamic Web Project w/Servlet

ITEM 1= Transitioning from Java EE to opensource Jakarta EE

READ THIS- <https://blogs.oracle.com/javamagazine/transition-from-java-ee-to-jakarta-ee>

The API JAR files had their Maven group ID changed from `javax.*` to `jakarta.*`,

ITEM 2= Install of Eclipse and setup project with A Jakarta Tomcat Container (for run time)based “Dynamic Web Project” Project in Eclipse

Install Eclipse IDE for Enterprise Java Developers (make sure you get the Enterprise Java Developer version).

>go to eclipse.org and find the LATEST version.

Example at time of creation of the document it was

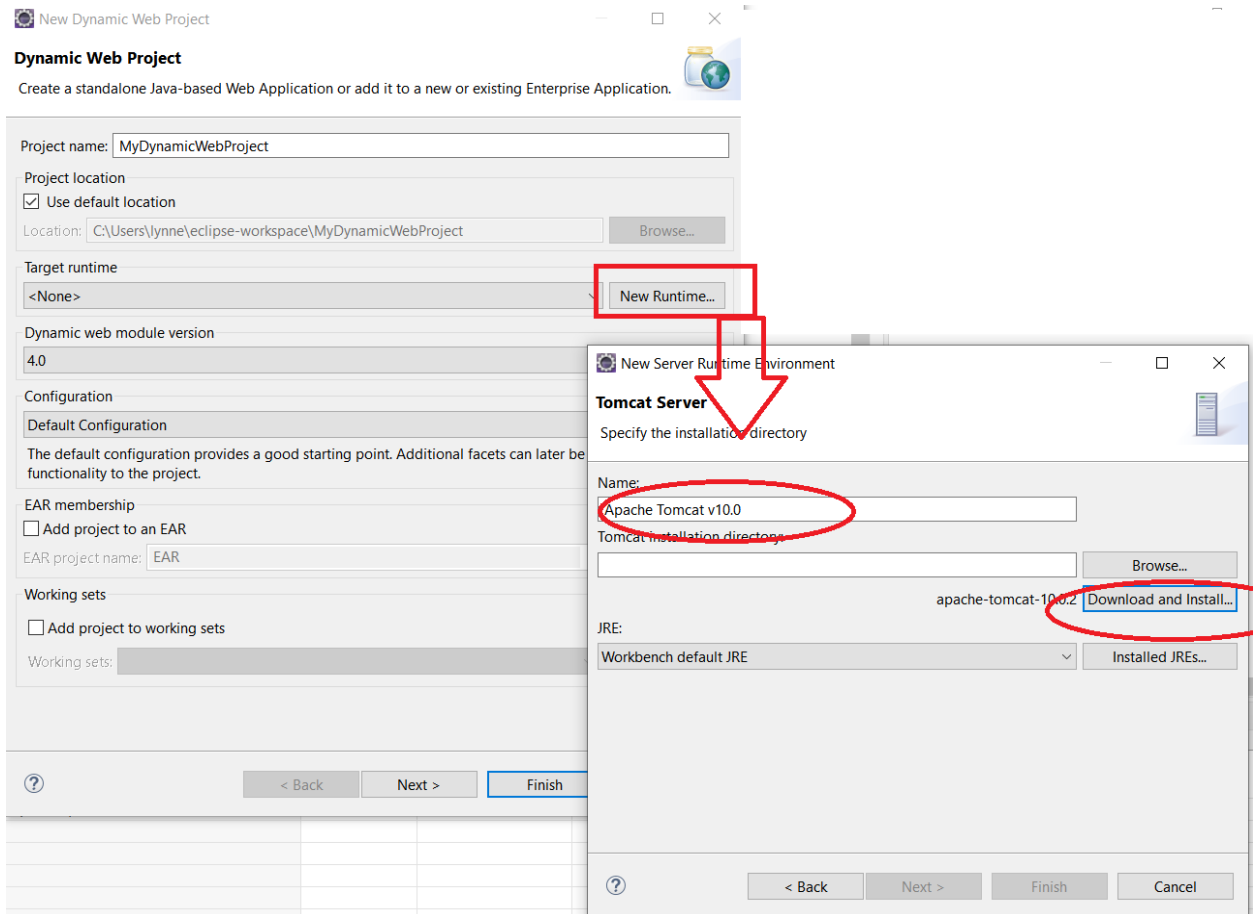
<https://www.eclipse.org/downloads/packages/release/2020-12/r/eclipse-ide-enterprise-java-developers>

ITEM 3= Create Dynamic Web Project with A Jakarta Tomcat Container (for run time)

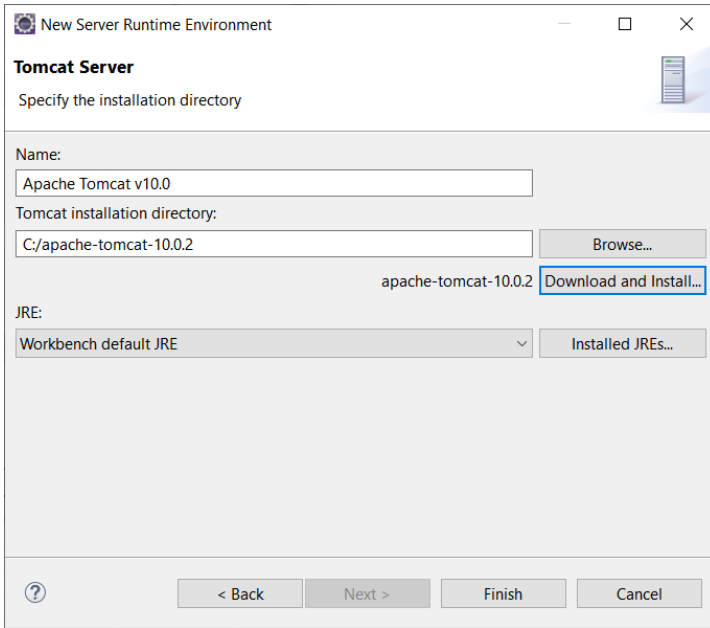
NOTE: first time it will prompt you to install Tomcat if you do not have it installed

- 1) File ->New Project-> Dynamic Web

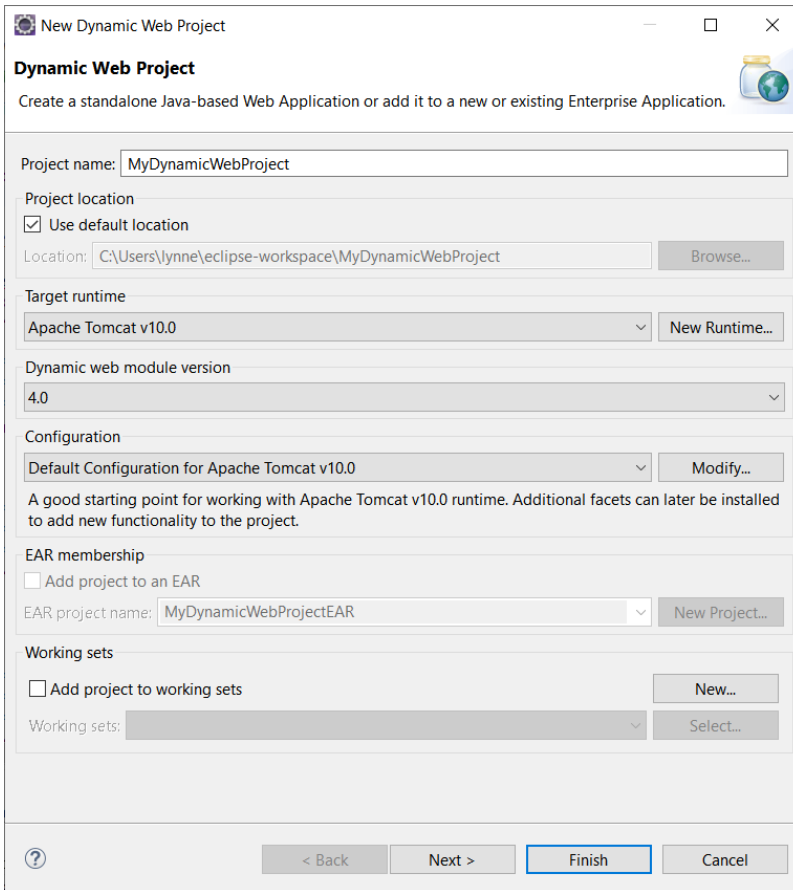
- 2) Specify name and select New Runtime (only need to do this first time)
Let's use a version of Apache Tomcat (a simple servlet container) and then you will need to either find the local location or do a download and install



Here I specified my C: drive for location to install



I will then get back the first frame



OPTION 1: Creation of Dynamic Web Application with No web.xml (uses modern Annotations) in Eclipse with Tomcat

- 1) File ->New-> Dynamic Web Project
specify the name and do not select web.xml

New Dynamic Web Project

Dynamic Web Project

Create a standalone Java-based Web Application or add it to a new or existing Enterprise Application.

Project name: MyTestDynamicWebProjectNOWebXML

Project location

Use default location

Location: C:\Users\lynn\workspace\MyTestDynamicWebProjectNOWebXML

Target runtime

Apache Tomcat v10.0

Dynamic web module version

4.0

Configuration

Default Configuration for Apache Tomcat v10.0

EAR membership

Add project to an EAR

Working sets

Add project to working sets

Finish

New Dynamic Web Project

Web Module

Configure web module settings.

Context root: MyTestDynamicWebProjectNOWebXML

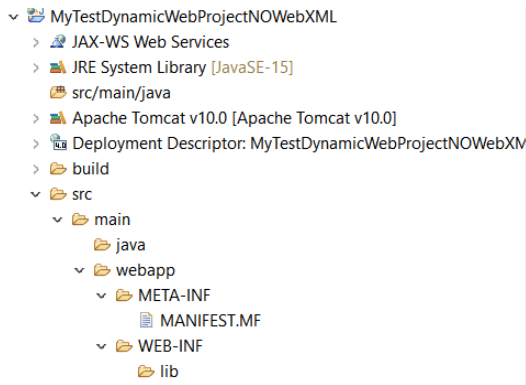
Content directory: src/main/webapp

Generate web.xml deployment descriptor

don't select this and will use more modern Java EE standard of annotations over web.xml specifications

Finish

Resulting structure...notice there is no Servlet, no JSP or HTML but, just an empty project structure waiting for YOU to add those things



2) Add a servlet (right click on src/main/java or on project and new -> other->servletor similar). Fill out the form with package name, servlet name and URL mapping(s) you wish.

Create Servlet
Specify class file destination.

Project: MyTestDynamicWebProjectNOWebXML

Source folder: /MyTestDynamicWebProjectNOWebXML/src/main/java

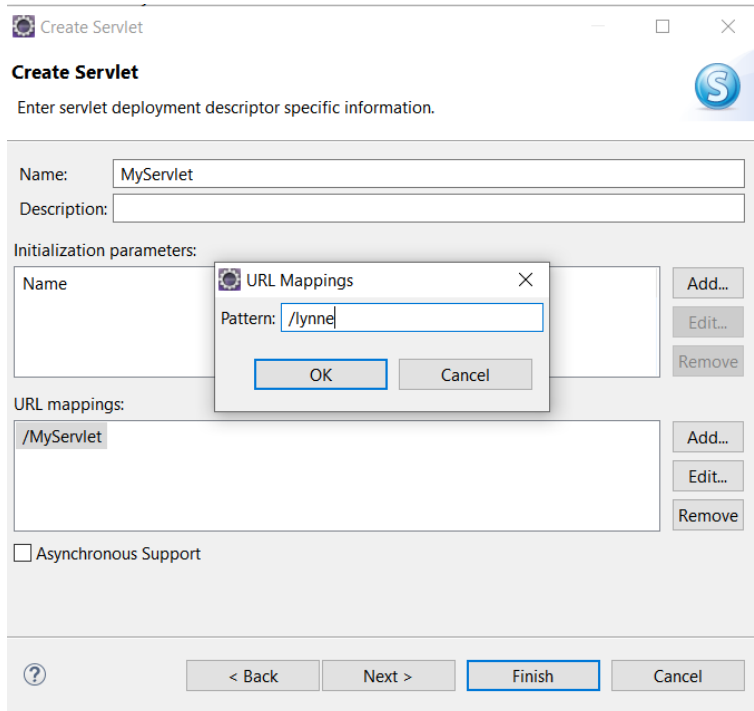
Java package:

Class name: MyServlet

Superclass: javax.servlet.http.HttpServlet

Use an existing Servlet class or JSP

Class name: MyServlet



- 3) NOW YOU need to make sure you are using jakarat packages and not javax (so I manually change this in the servlet code). ---I am not sure why it defaults to the javax (older Java EE) rather than the newer jarkata EE package or how to fix this.

```

1
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import jakarta.servlet.ServletException;
6 import javax.servlet.annotation.WebServlet;
7 import jakarta.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import jakarta.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import jakarta.servlet.http.HttpServletRequest;
12 import javax.servlet.http.HttpServletResponse;
13 import jakarta.servlet.http.HttpServletResponse;
14
15 /**
16  * Servlet implementation class MyServlet
17  */

```

*Tomcat uses jakarta and nto javax for the last ** versions of Tomcat*

SO YOU MUST replace the imports with the equivalent in the jakarta package

.....here is the result of my doing this. Notice the web annotation @WebServlet and here I specified 2 URL mappings that will work

```

import java.io.IOException;
import jakarta.servlet.*;
import jakarta.servlet.annotation.*;
import jakarta.servlet.http.*;

```

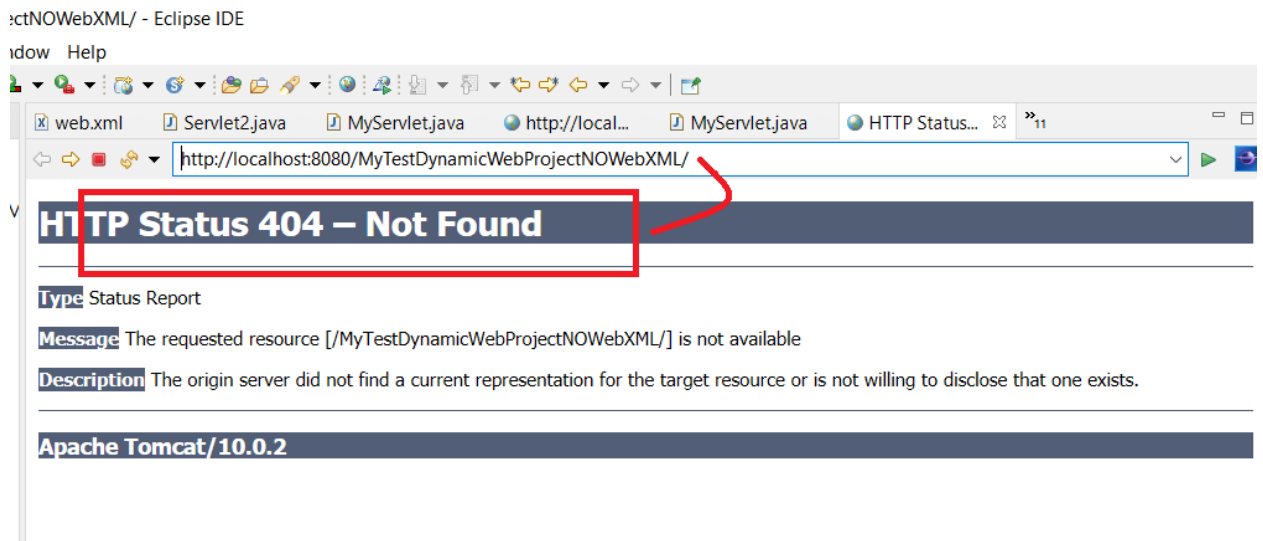
```

/*
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 MyServlet
 */
@WebServlet({ "/MyServlet", "/lynne" })
public class MyServlet extends HttpServlet {

```

- 4) Run by right clicking on project and say run as. However, typically you will add index.html or jsp pages and those will be run instead of directly as server. to your dynamic web applications...but, that is another discussion.

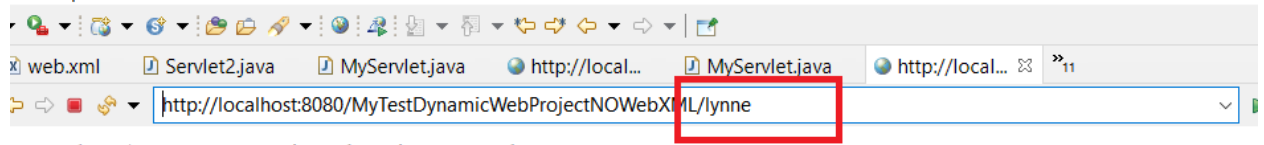


NOTE as there is no index.* file and hence we get a 404...change the url you see here and append to it the web annotation endpoint. For example, I

will add /lynne to the end of the url and now it will invoke my servlet

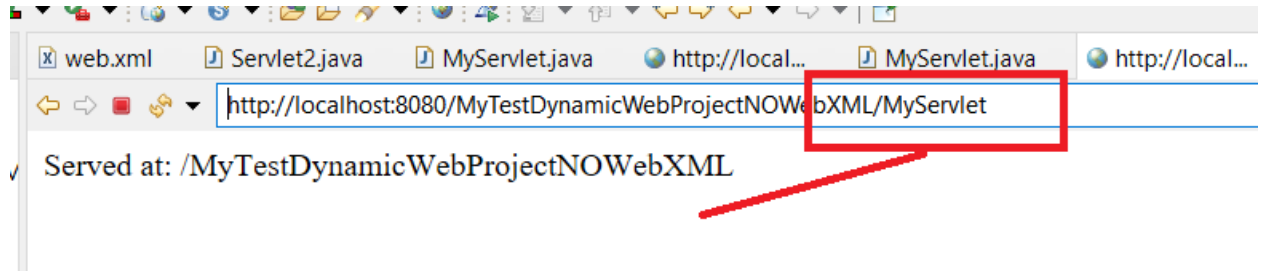
MyTestDynamicWebProjectNOWebXML - Eclipse IDE

w Help



Served at: /MyTestDynamicWebProjectNOWebXML

Because I have TWO url mappings specified in the annotation @WebServlet in my Servlet java code, I can use either of them to invoke the servlet. Here you can see the send running



Served at: /MyTestDynamicWebProjectNOWebXML

OPTION 2: Creation of Dynamic Web Application with web.xml in Eclipse with Tomcat

- 1) Step 1-3 are the same except you select the web.xml creation box.
- 2) Edit the web.xml file so that the Desired URL mappings for your new servlet are setup.

...here is an example for a servlet I created called LynneServlet that is mapped to /lynne and then you can see me running it and the corresponding url

The screenshot displays the Eclipse IDE interface. The top toolbar includes icons for Edit, Source, Navigate, Search, Project, Run, Sapphire Samples, Window, and Help. The Project Explorer on the left shows the project structure: MyTomcatDynamicWebProject, Deployment Descriptor: MyTomcatDynamicWebProject, JAX-WS Web Services, JRE System Library [JavaSE-15], src/main/java, Apache Tomcat v10.0 [Apache Tomcat v10.0], build, src, main, java (containing LynneServlet.java), webapp (containing META-INF with MANIFEST.MF and WEB-INF with lib and web.xml), and LynneJSP.jsp. The main editor shows the web.xml file with the following XML content:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <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_3_0.xsd" >
3   <display-name>MyTomcatDynamicWebProject</display-name>
4   <welcome-file-list>
5     <welcome-file>index.html</welcome-file>
6     <welcome-file>index.htm</welcome-file>
7     <welcome-file>index.jsp</welcome-file>
8     <welcome-file>default.html</welcome-file>
9     <welcome-file>default.htm</welcome-file>
10    <welcome-file>default.jsp</welcome-file>
11  </welcome-file-list>
12
13  <servlet>
14    <servlet-name>LynneServlet</servlet-name>
15    <servlet-class>LynneServlet</servlet-class>
16  </servlet>
17
18  <servlet-mapping>
19    <servlet-name>LynneServlet</servlet-name>
20    <url-pattern>/lynne</url-pattern>
21  </servlet-mapping>
22 </web-app>
```

Red boxes highlight the web.xml file in the Project Explorer and the servlet-mapping configuration in the code editor. A red arrow points from the web.xml file in the Project Explorer to the servlet-mapping configuration in the code editor. Below the code editor, a browser window shows the URL `http://localhost:8080/MyTomcatDynamicWebProject/lynne` and the message "Served at: /MyTomcatDynamicWebProject".

ITEM 4= Adding html and jsp pages to your project

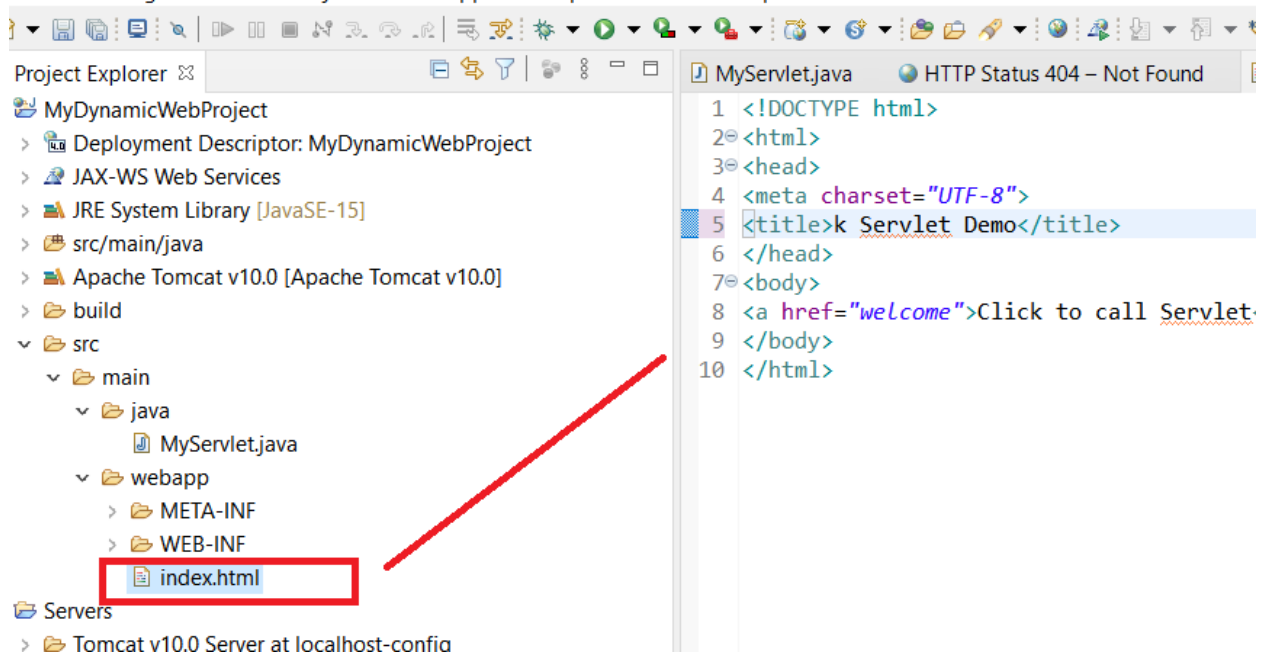
Create an html page to call the servlet class on a webpage

We are creating an html file that would call the servlet once we click on the link on web page. Create this file in WebContent folder. The path of the file should look like this: WebContent/index.html

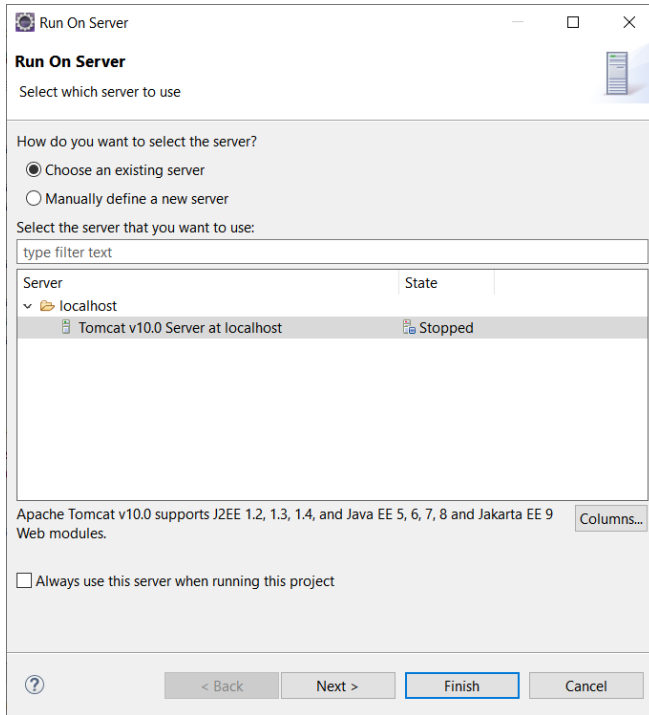
index.html

NOW lets create our index.html page that will invoke our sevlet. Right click on web folder and create NEW html file

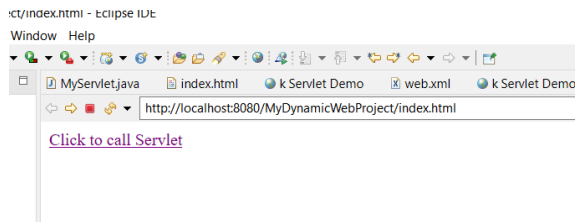
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Servlet Demo</title>
</head>
<body>
<a href="welcome">Click to call Servlet</a>
</body>
</html>
```



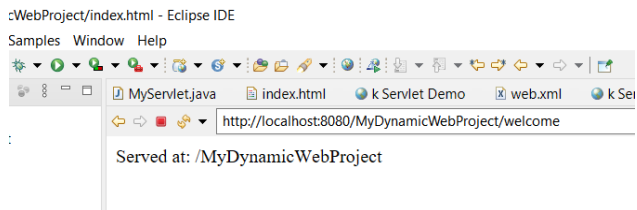
NOW click on Run on your tomcat server RIGHT click on index.html -> run on server and select your tomcat server



Here is the result of running index.html



AND clicking on the link I will get my servlet



NOTE you can add JSP file in the same fashion. Both JSP and HTML files will put under webapp directory and both can be directly run

The screenshot displays the Eclipse IDE interface. The top menu bar includes File, Edit, Navigate, Search, Project, Run, Sapphire Samples, Window, and Help. The Project Explorer on the left shows two projects: MyDynamicWebProject and MyTomcatDynamicWebProject. The MyTomcatDynamicWebProject structure is expanded to show the webapp directory containing META-INF, WEB-INF, lib, web.xml, and LynneJSP.jsp. A red box highlights the web.xml and LynneJSP.jsp files, with a red arrow pointing from this box to the browser window. The browser window shows the URL http://localhost:8080/MyTomcatDynamicWebProject/LynneJSP.jsp and the output text "hi lynne this is static so far". The bottom console shows the Tomcat v10.0 Server at localhost [Started, Synchronized].

ITEM 4: miscellaneous...tips, etc.

MAKE SURE your project has the Project Properties -> Java Build Path- and it contains the Apache Tomcat (hit apply and close)

