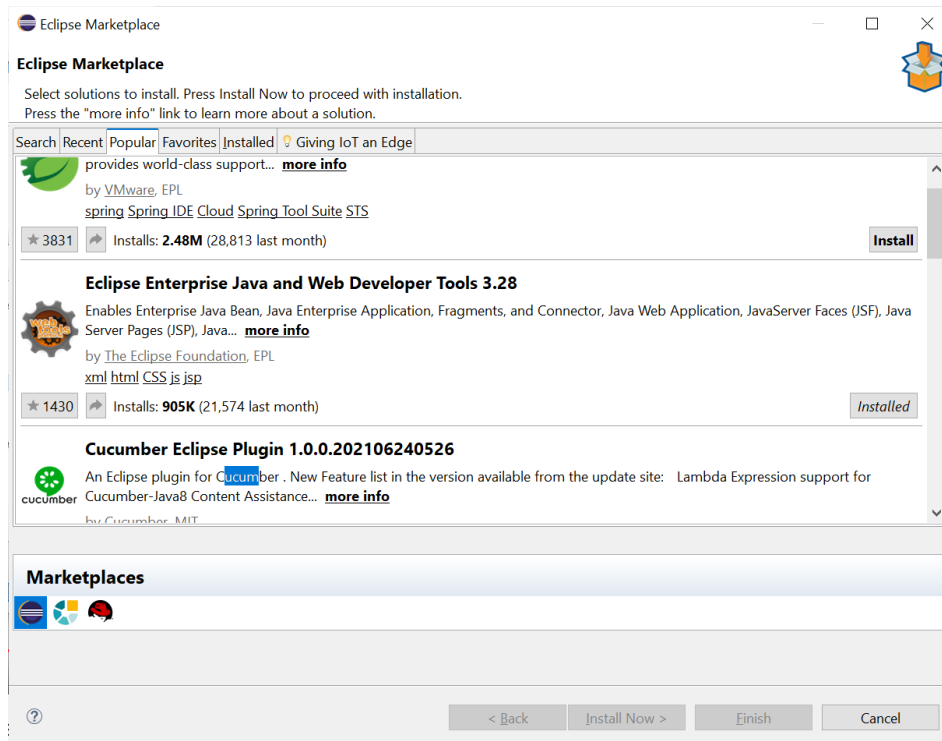


## Configuring JEE working environment in Eclipse

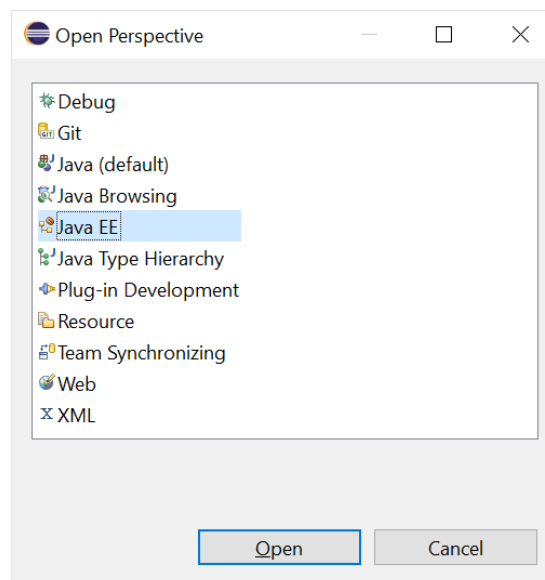
### Install JEE development tools in Eclipse

Run Eclipse and go to menu **Help-> Eclipse Marketplace**.

Find and install: **Eclipse Enterprise Java and Web Developer Tools**.

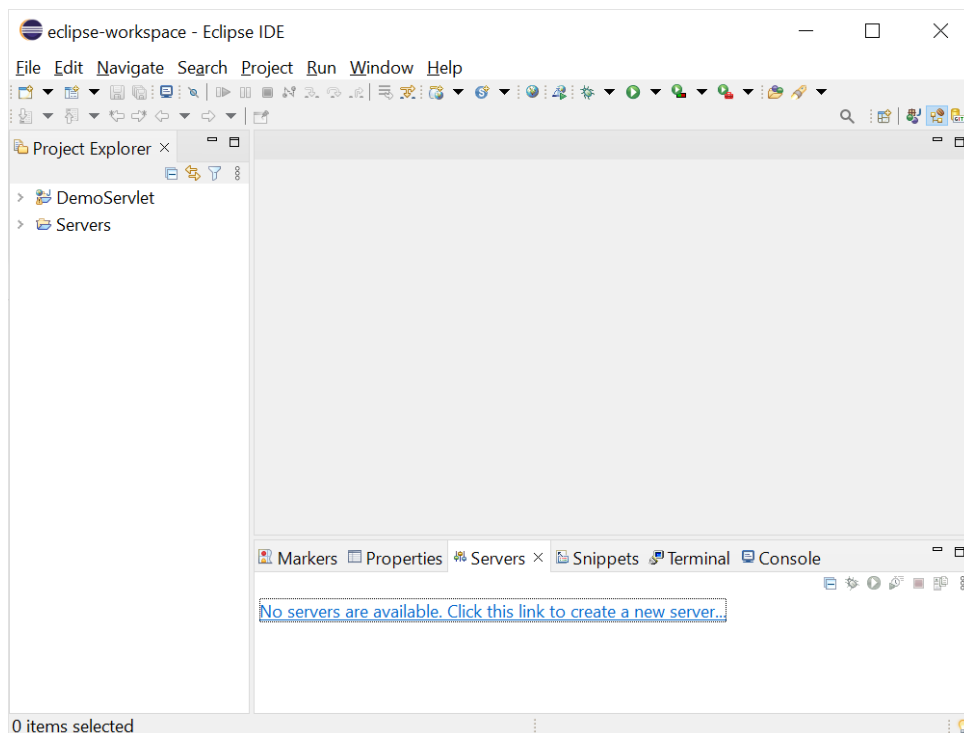


After installation, switch Eclipse working environment to **JEE Perspective** that allows development of distributed enterprise applications with JEE technologies. In left upper corner of Eclipse there is a button for selecting perspective (Open Perspective). Just select **Java EE**.

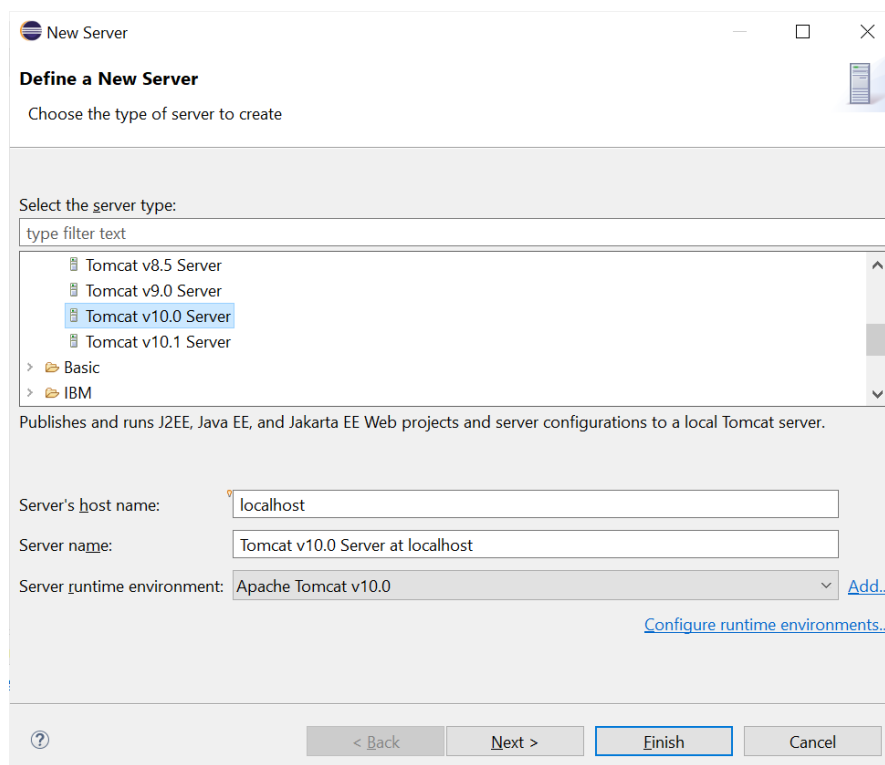


## Install and configure Tomcat server within Eclipse

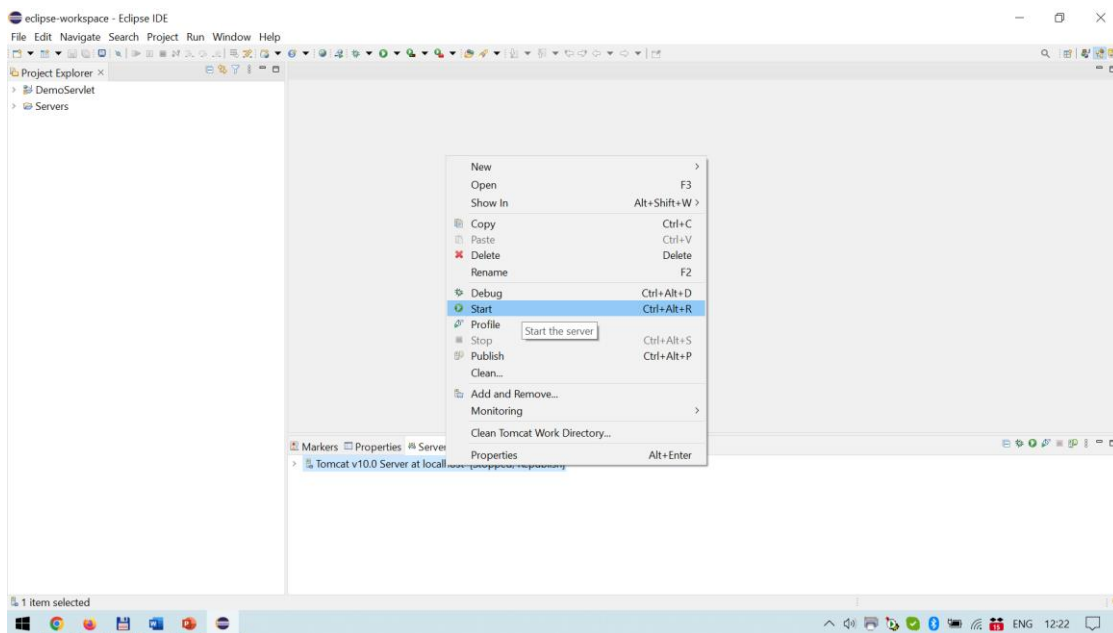
The next step is to add Tomcat Server to **Servers** tab in Eclipse.



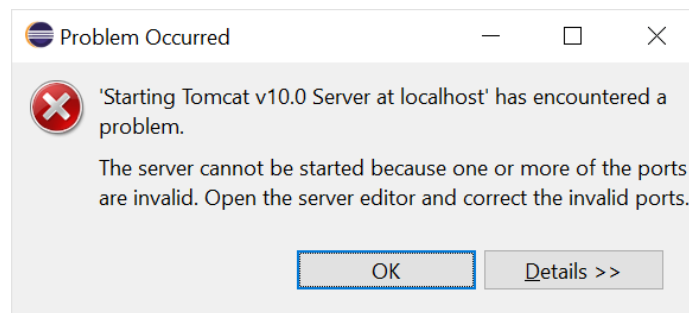
Just click the link to add Tomcat server to Eclipse JEE environment. Select Tomcat version and install server with the name **localhost**, and in the next step select folder where Tomcat is installed.



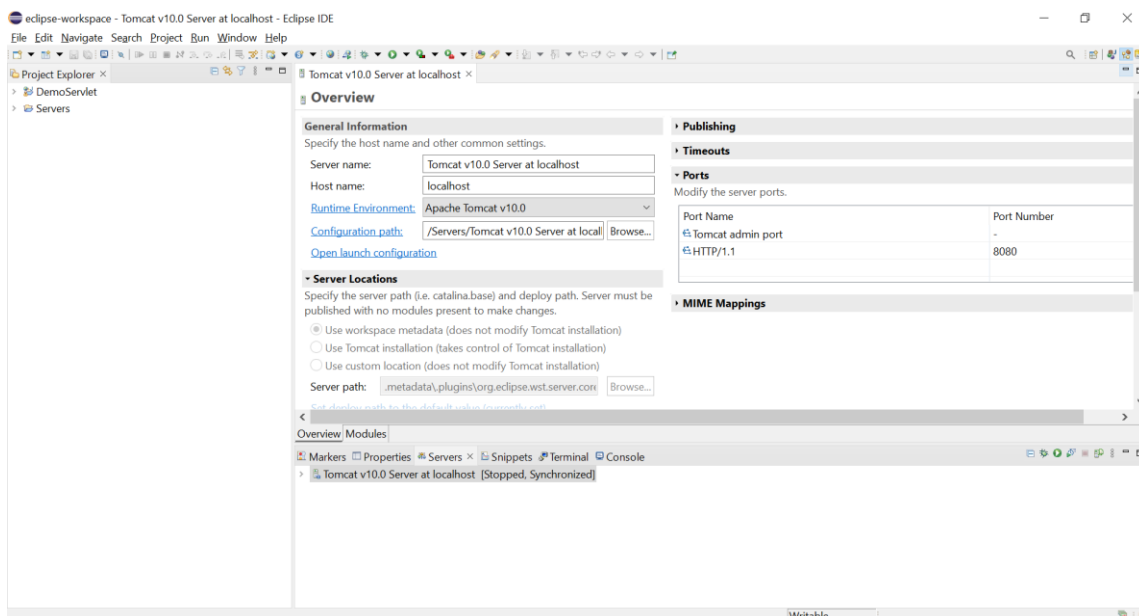
Installed server is stopped, so it should be started (right mouse click on server).



Typical problem with starting servers is with invalid ports, as it is stated with the following error message.

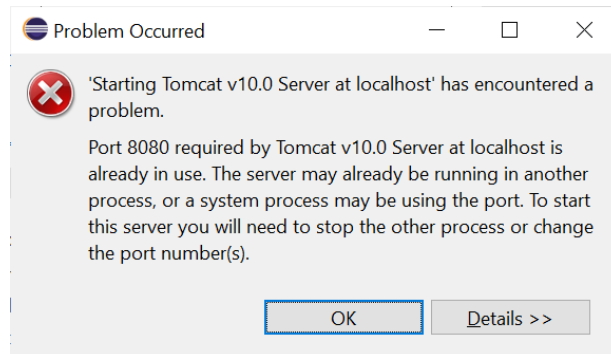


To solve this problem just double click on Tomcat Server to change configuration.

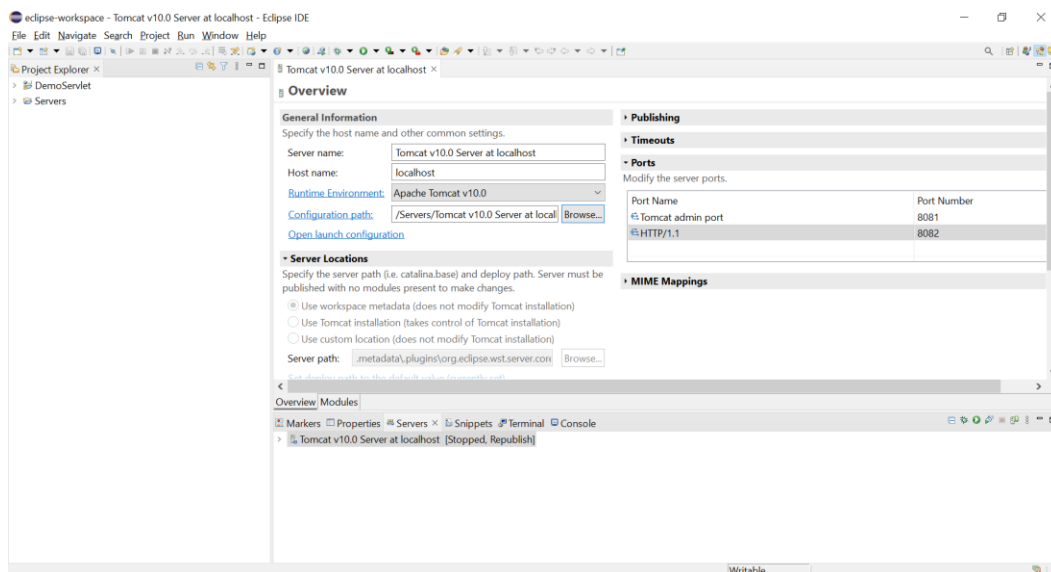


Set **Tomcat Admin Port** to 8081, and save configuration. Try to starting again.

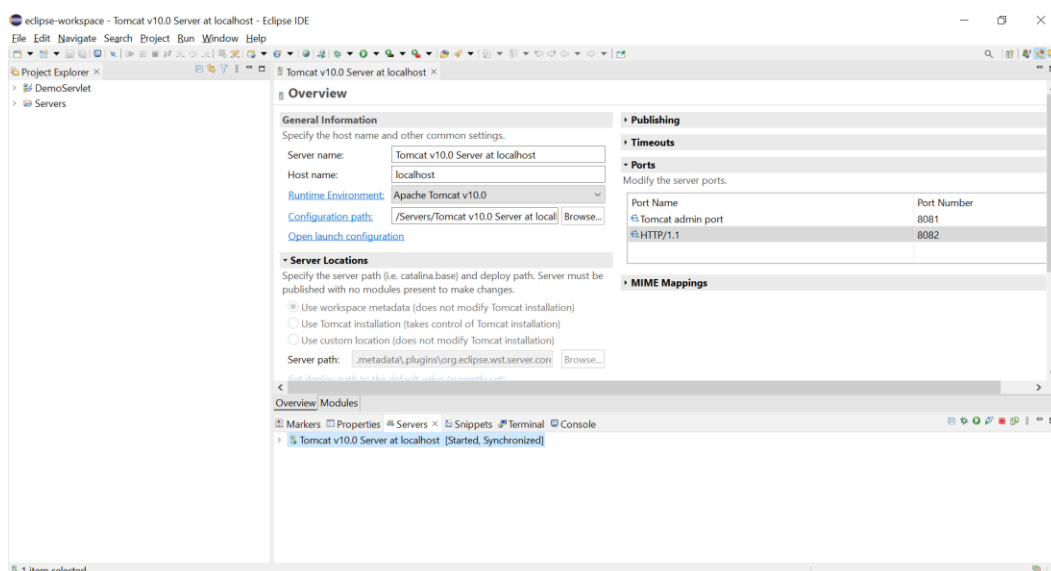
The next problem relates to used port 8080.



Just change **HTTP port** from 8080 to 8082 and save new configuration.



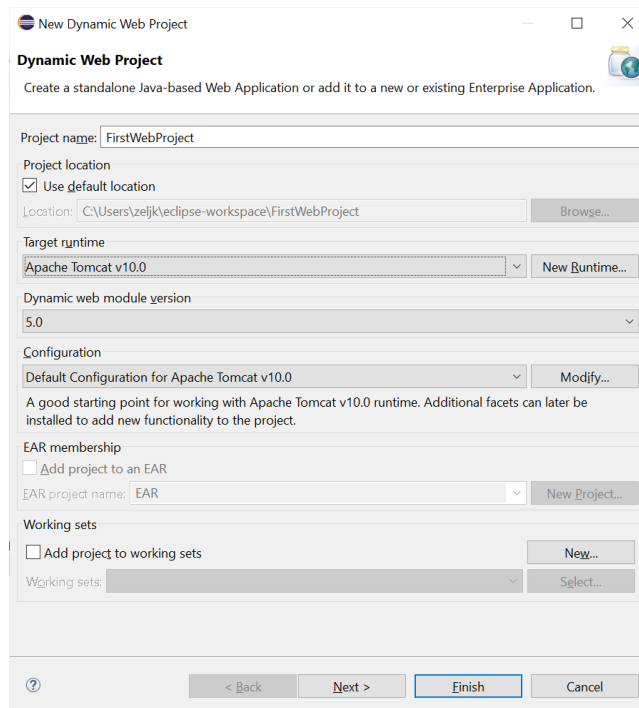
Try starting Tomcat again [successful!]. Tomcat is started and synchronized.



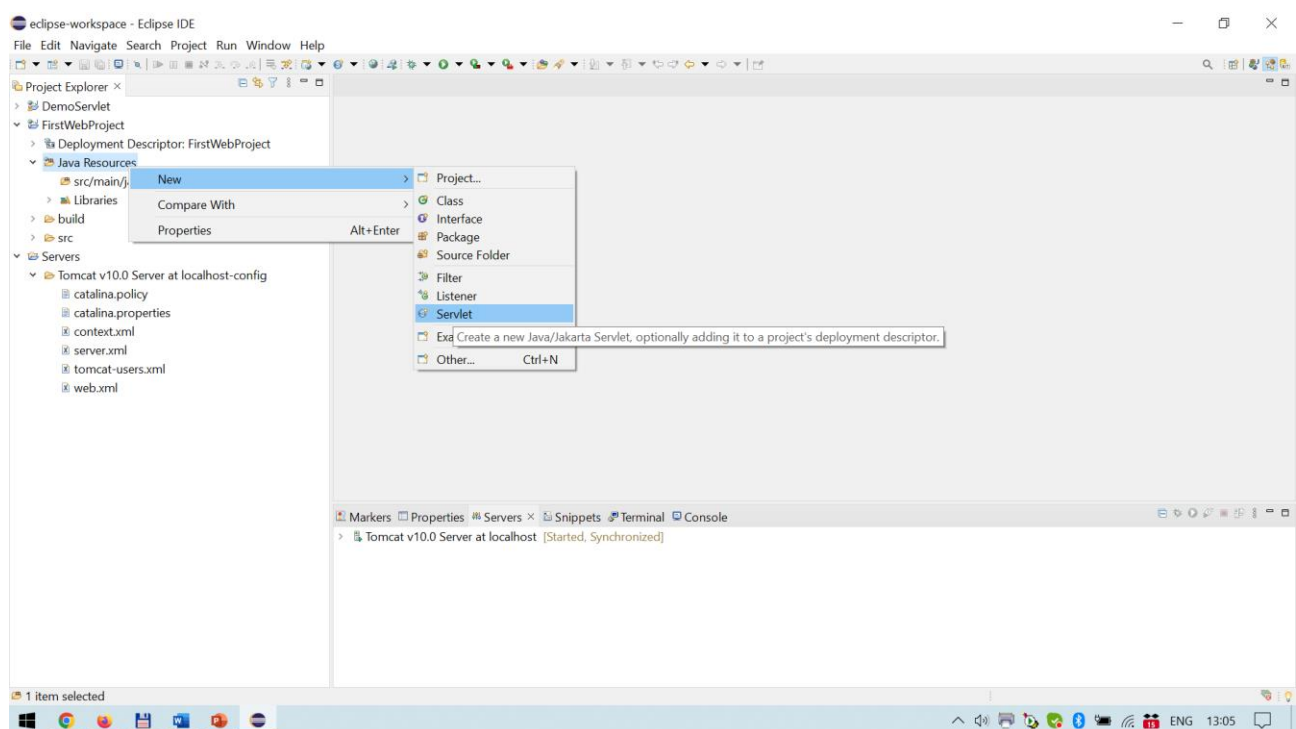
## Create the first web project

Select menu option: **File -> New -> Dynamic web project**

Add project name and select **Target runtime** to be installed **Apache Tomcat**.



Add first servlet to the project.



Set servlet name and package in dialog **Create Servlet**, and do not change other parameters when moving to the next steps.

Now, in Eclipse we have Jakarta EE, which is the same as Java EE (JEE) that is legally owned by Oracle. The transition was in 2018.

So, we have new classes for working with Servlets in package **jakarta**:

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
```

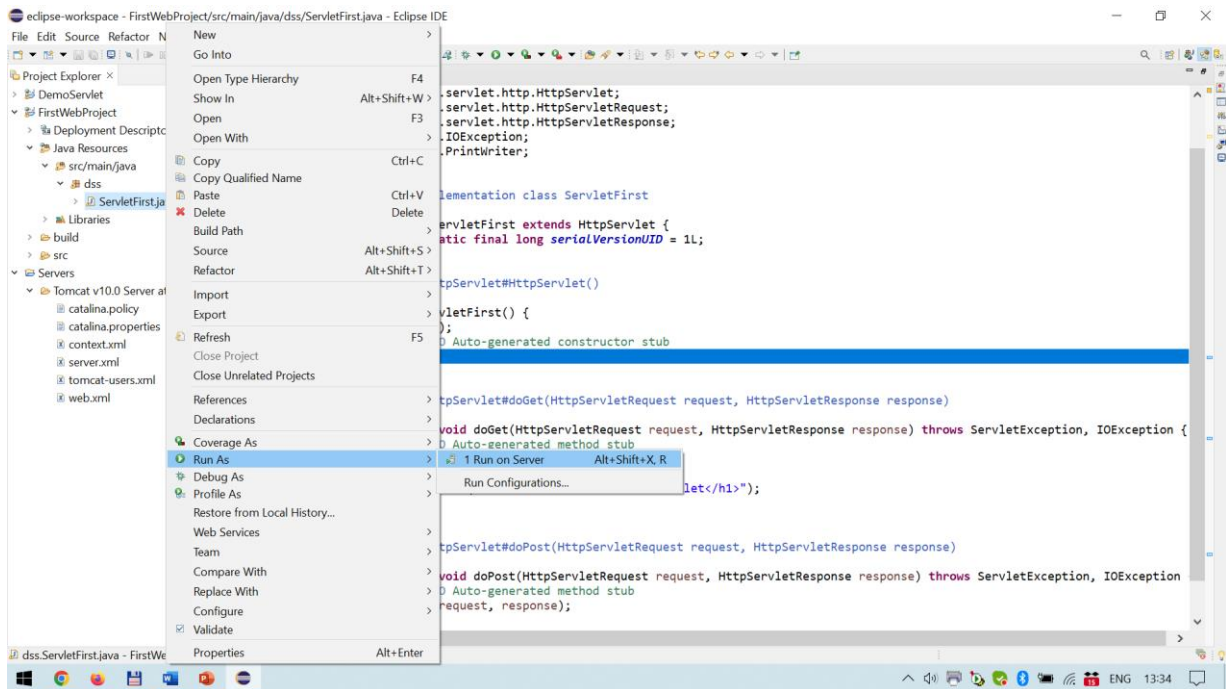
Just to compare with older versions of JEE and Tomcat that supported Servlet classes in package **javax**.

```
import javax.servlet.*;
import javax.servlet.http.*;
```

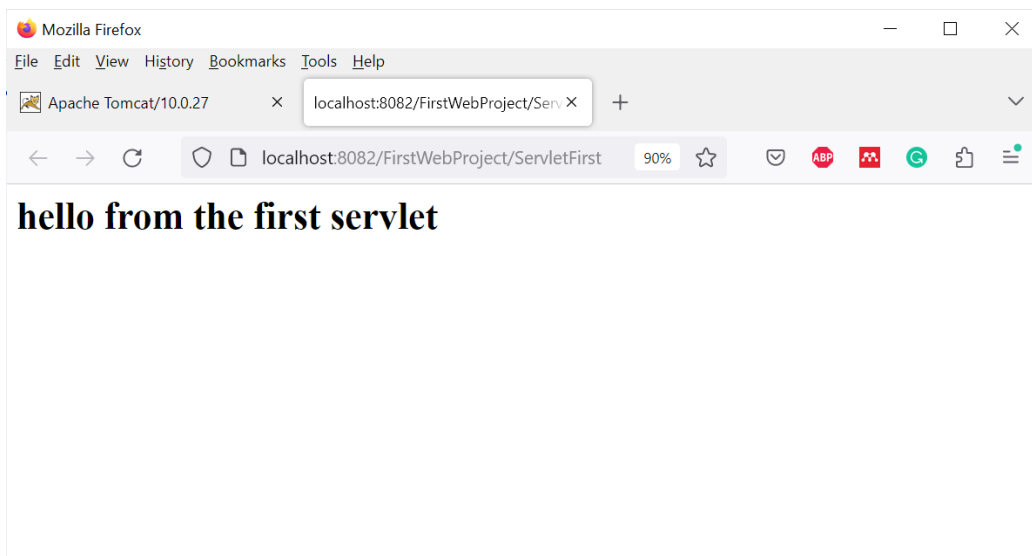
Some modifications to `doGet()` method to enable servlet to work (print message):

```
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    PrintWriter pw = response.getWriter();
    response.setContentType("text/html");
    pw.println("<h1>hello from the first servlet</h1>");
}
```

## Running servlet from the Eclipse workspace



Execution in web browser.



Elements of execution:

Server name and port: **localhost:8082**

Web application: **FirstWebProject**

Servlet: **ServletFirst**