







Web Development with Eclipse and Tomcat

This guide is for setting up [Eclipse](http://www.eclipse.org/)  and [Tomcat](http://tomcat.apache.org/)  for Java web application development on Windows 7/8/10. The process on MacOS and Linux should be quite similar. A video showing the process is available on [YouTube](https://www.youtube.com/watch?v=rr44QRtAVKU)  (<https://www.youtube.com/watch?v=rr44QRtAVKU>).

Tomcat


Download the latest Tomcat 9 binary release from the [Apache Tomcat Project](https://tomcat.apache.org/download-90.cgi)  (<https://tomcat.apache.org/download-90.cgi>). Note that there are several packages available for download. You should download the ZIP file under the "Core" distribution. After downloading the file, unzip it to a local directory, e.g. `c:\apache-tomcat-9.0.x`.

Add the jar files of the following libraries to the `lib` folder under the Tomcat directory:

- `jstl.jar` and `standard.jar` from [JSP Standard Tag Library \(JSTL\)](https://csns.cysun.org/download.html?fileId=3812279)  (<https://csns.cysun.org/download.html?fileId=3812279>).
- `mysql-connector-java-<version>.jar` from [MySQL JDBC Driver](http://www.mysql.com/downloads/connector/j/)  (<http://www.mysql.com/downloads/connector/j/>) (in the drop-down list, select Platform Independent and download the ZIP file).

A common mistake is to just copy the ZIP files to the `lib` folder - please don't do that. Please unzip the ZIP files and copy the JAR files to the Tomcat `lib` folder.

Eclipse

Download the latest **Eclipse IDE for Enterprise Java and Web Developers** from [Eclipse.org](https://www.eclipse.org/downloads/packages/)  (<https://www.eclipse.org/downloads/packages/>). A common mistake here is to download the one for "Java Developers" - note that we need the one for "Enterprise Java and Web Developers". If you used Eclipse for other classes, chances are it was the one for Java Developers, and you'll need to download the one for Enterprise Java and Web Developers for this class.

There is now an Eclipse Installer, but personally I prefer the zip package as it does not require internet access or admin privilege during installation. Unzip it to a local directory, e.g. `c:\eclipse`, then start up Eclipse by double-click on `eclipse.exe` under the Eclipse folder. If you are using Eclipse for the first time, you will be asked to choose a folder to be Eclipse's workspace. Eclipse will store all your projects in that folder.

Please read [this article \(https://www.genuitec.com/stop-slow-eclipse-myeclipse-startups/\)](https://www.genuitec.com/stop-slow-eclipse-myeclipse-startups/) on how to fix Eclipse slow startup on Windows 10. The TLDR version is that you should exclude the Eclipse folder and the workspace folder from Windows Defender Antivirus scan - this will vastly improve Eclipse's startup speed on Windows 10.

Sample Application

We can now create a sample web application using Eclipse. The application consists of a servlet called `HelloServlet` and a JSP page `HelloJSTL.jsp`.

First, in Eclipse, create a Dynamic Web Project (File -> New -> Project ... -> Dynamic Web Project). You need to specify the project name and a target runtime. We'll use `webtest` as the project name for this sample application. As for target runtime, if this is the first time you create a Dynamic Web Project, the Target runtime dropdown list will show `<None>`. You can click the New Runtime button to create a new one: select Apache Tomcat v9.0 as the new server runtime environment, then on the next screen, enter the path of your Tomcat directory for the installation directory.

After the Target runtime is created, continue with the project creation wizard. You may leave all options to default except that you should check the option `Generate web.xml deployment descriptor`.

Create a new servlet (Right click on the project name -> New -> Servlet). Enter `webtest.servlet` as the package name and `HelloServlet` as the class name then click Finish. Edit `HelloServlet.java` so the `doGet()` method looks like the following:

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.getWriter().print("Hello Servlet");
}
```

Create a new JSP page (Right click on the project name -> New -> JSP). Enter `HelloJSTL.jsp` as the file name then click Finish. Edit `HelloJSTL.jsp` so it looks like the following:

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>HelloJSTL</title>
</head>
<body>
    <c:out value="Hello JSTL" />
</body>
</html>
```

Run the application (Right click on the project name -> Run As -> Run on Server). Select Tomcat v9.0 Server as the server type then click Finish. Enter the URL `http://localhost:8080/webtest/HelloServlet` in a browser and you should see "Hello

Servlet" output by the servlet, and enter the URL

`http://localhost:8080/webtest/HelloJSTL.jsp` and you should see "Hello JSTL" displayed by the JSP page.

Deployment on CS3

The JDK on CS3 is version 11. If your JDK version is higher than 11, which would be the case if you use a newly downloaded Eclipse, you will see an error like [this](#)

(<https://calstatela.instructure.com/courses/80633/files/11500711?wrap=1>)_ ↓

(https://calstatela.instructure.com/courses/80633/files/11500711/download?download_frd=1) when you deploy your application on CS3. You must configure Eclipse to generate JDK 11 class files as follows:

- Right click on the project and select Properties -> Project Facets
- For the Java project facet, select 11 from the Version drop-down list
- Click Apply and Close

Unfortunately you have to remember to do this for every new project created in Eclipse. *(If you know how to change the default Java facet version, please let me know.)*

To deploy your application onto the CS3 server, you simply need to transfer the following files from your Eclipse project to their corresponding folders under your CS3 account:

- All the files and subfolders under the project `build/classes` folder should be transferred to the `www/WEB-INF/classes` folder under your CS3 account.
- All the files and subfolders under the project `src/main/webapp` folder, excluding the `META-INF` subfolder, should be transferred to the `www` folder under your CS3 account.
- Transfer the `web.xml` file to the `www/WEB-INF` folder under your CS3 account. This should be done after all the other files are transferred, and later whenever you update your class files on CS3, you must re-upload the `web.xml` file even though its content is not changed. This is because the Tomcat server monitors the `web.xml` file of each project, and uploading/re-uploading `web.xml` basically tells the Tomcat server to load/reload your project.

After your application is deployed on the CS3 server, you may access it using the following URL:

```
http://cs3.calstatela.edu:8080/<username>/<servlet_or_jsp_name>
```

where `<username>` is your account name on the server.