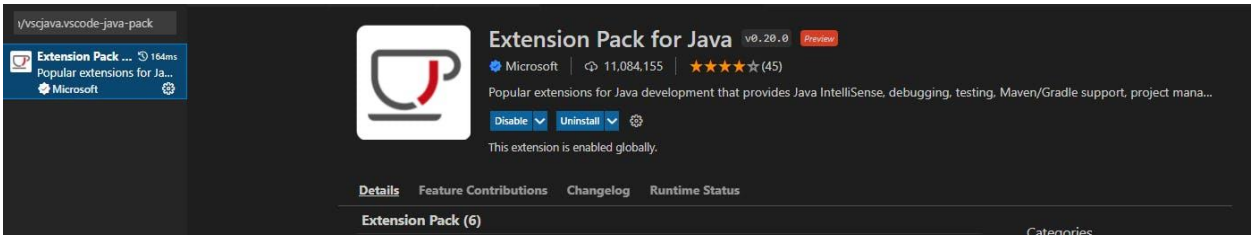
How to create Java Servlet with Tomcat and Maven in VS Code.

Setups and Configurations

First we need to Download and install Java 11, Apache Maven, Tomcat 10 Server and Visual Studio Code(VSC) IDE into your system. Make sure to follow the proper procedural on installing all the mentioned technologies.

After the Installation of the above we need to set working platform by adding some extensions to VSC that will assist us in creating out project.

Let's Install the Visual Studio extension pack for java in the VSC by hold (CTRL + SHIFT + X) inside VSC and
type `vscode:extension/vscjava.vscode-java-pack`.



Create Maven Project

Now is time to create a maven project, create a folder to your desired location, which will be used for your project.

From the Command Palette (Crtl+Shift+P) in Visual Studio Code, then

- Select `Maven: Create Maven Project`

- Select `maven-archetype-quickstart`
- Select the current version like `1.4`
- Type your groupId like `com.myservlet`
- Type your artifactId like `servlet`
- Locate your destination folder which you created before, then generate the project there.

Now it is time to compile our project with maven to make sure everything is okay. Click on `Terminal` on top bar then click `New Terminal` and then execute command `mvn compile` .The project will be complied with maven.

Add Dependencies

Before we start, code our servlet we need to add the important dependencies for project in pom.xml file. Without these dependencies, we will not be able to build our servlet.

```
<dependency>
            <groupId>jakarta.platform</groupId>
            <artifactId>jakarta.jakartaee-api</artifactId>
            <version>9.0.0</version>
            <scope>provided</scope>
 </dependency>
```

Create Servlet

- Now we are ready to start creating our servlet, we will start by deleting test package and the default `App.java` file generated by the IDE inside directory `src\main\java\com\myservlet\App.java`.
- Then we will create a new simple **java class** called `HelloServlet.java` inside directory `src\main\java\com\myservlet\HelloServlet.java`. The class will implement `Servlet` interface.
- In order for our Servlet to work we need a **mapping file** for the requests to the **Servlet**, the **mapping file** in Servlet technology is called `web.xml`.

Create a web.xml file

- Create a folder `webapp` under directory "src/main/`webapp`"
- Create a folder `WEB-INF` under directory "src/main/webapp/`WEB-INF`",then
- Create a file `web.xml` under directory "src/main/webapp/WEB-INF/`web.xml`"

NB: Make sure the `web.xml` file is properly allocated as above.

Add servlet configuration in the `web.xml`.

```
<web-app xmlns= "https://jakarta.ee/xml/ns/jakartaee"
 xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation= "https://jakarta.ee/xml/ns/jakartaee
                      https://jakarta.ee/xml/ns/jakartaee/web-app_5_0.xsd"
 version= "5.0"
 metadata-complete= "true">
      <servlet>
            <servlet-name>Testing</servlet-name>
            <servlet-class>MyFirstServlet</servlet-class>
      </servlet>
      <servlet-mapping>
```

```
        <servlet-name>Testing</servlet-name>
        <url-pattern>/MyTest</url-pattern>
    </servlet-mapping>
</web-app>
```

Generate a war file

Now we need to generate a war file to test our project in a server. But before that we need to make sure the `<packaging>war</packaging>` in `pom.xml` file of your project is **war** like below and not **jar**.

```
<groupId>com.myservlet</groupId>

<artifactId>servlet</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>war</packaging>
```

Time to generate our `war` file, compile the project by click on `Terminal` on top bar of Visual Studio Code then click `New Terminal` and then execute command `mvn compile`.

Deploy war file to local tomcat server

We will use maven plugin to create war and deploy it on our local **tomcat server**.

To do this, add the following plugins

- This one for compiling

```
<plugin>

    <groupId>org.apache.maven.plugins</groupId>
```

```
    <artifactId>maven-compiler-plugin</artifactId>
     <version>3.8.1</version>
    <configuration>
          <source>11</source>
          <target>11</target>
    </configuration>
</plugin>
```

- This one for generating war file

```
<plugin>
     <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-war-plugin</artifactId>
    <version>3.3.0</version>
    <configuration>
          <failOnMissingWebXml>false</failOnMissingWebXml>
    </configuration>
 </plugin>
```

- And you can use this plugin for deploying:

```
<plugin>
     <groupId>org.apache.tomcat.maven</groupId>
     <artifactId>tomcat7-maven-plugin</artifactId>
     <version>2.2</version>
     <configuration>
```

```
            <username>admin</username>

            <password>admin</password>

            <url>http://localhost:9090/manager/text</url>

            <path>/hello</path>

        </configuration>

</plugin>
```

Now we need to configure your server. After downloading **tomcat server,** open server.xml file located in conf folder in your tomcat directory and change the startup port

```
<Connector port="9090" protocol="HTTP/1.1"

                connectionTimeout="20000"

                redirectPort="8443" />
```

Then, update the tomcat-users.xml by adding the following roles and users

```
<role rolename="tomcat"/>

    <user password="tomcat" roles="tomcat"
username="tomcat"/>

<user password="admin" roles="manager-gui,manager-
script,manager-jmx,manager-status,admin" username="admin"/>
```

Start your server from bin folder by opening cmd and execute startup.bat.

Now you are ready to deploy and run your Servlet app. Get back to VS-Code and then execute command `mvn install tomcat7:deploy` . Call your Servlet from the Web Browser

http://localhost:9090/test/MyTest