

# DO'S, ERRORS AND FIX\_SERVLET-DB-JSP-MAVEN\_ECLIPSE

## Common Errors and their Solutions in Servlet-Database Program

1. **ClassNotFoundException** - Database driver jar file missing from Project's WebContent/WEB-INF/lib folder
2. **Redirecting to login page again instead of another :-**
  - a. forgot 'commit' in db
  - b. wrong sql query string
3. **If getting error http status 404 or 405 (Get or Post Method Not Allowed) :-**
  - Check if method mentioned in form tag of html file and the method defined in servlet java file is same or not.
  - method="get" OR no method mentioned: doGet() in servlet java file
  - method="post" : doPost() in servlet java file
4. **For Tomcat Server or its version issue, try b. if a. doesn't fix the errors**
  - a. right click on Project -> Properties -> Targeted Runtimes -> select Apache Tomcat -> Apply and Close
  - b. right click on Project -> Build Path -> Configure Build Path -> Libraries tab -> Add Library -> Server Runtime -> Select Tomcat version you downloaded in your system -> Finish
5. **If errors in all files and server not starting, maybe the tomcat folder downloaded is corrupted.**
  - Download tomcat zip folder again.
6. **Kuch samajh ni aa rha :-**
  - a. clean tomcat server build, restart server and run html file again
  - b. If a. doesn't work, close eclipse and laptop, and sleep for a while. Then try again later or ask sir, the next day.

## Common Do's, Errors and their Solutions in JSP-Servlet Program

1. **Error 500 :-**
  - a. run servlet class file not jsp file.
  - b. take care of <%= %> '=' sign while writing java code in jsp file wherever required.
2. **DO** - ctrl+space while writing java code in jsp...to import corresponding classes and packages.
3. **DO** - Build Automatically must be enabled in Project.
4. **DO** - Select Java version 8 while creating the Dynamic Web Project.
5. **For java version issue or jre library issue,**
  - right click on Project -> Build Path -> Configure Build Path -> Libraries tab -> remove the existing jre library and click on Add Library -> JRE system library -> Select java 8 -> Finish.
6. **TIP** - If project shows a cross error, search 'Problems' in eclipse and check the list of errors displayed in problems window. Fix the errors accordingly.
7. Corrupted workspace could be an issue too sometimes due to similar name of workspace and projects inside it.
8. **If everything seems correct and still no output :-**
  - Create a new project and start again. Amen.

## Common Do's, Errors and their Solutions in Maven-DB-JSP-Servlet Program

We have created a MAVEN PROJECT (ci-webapp : class example);

Perspective - JAVA EE

**Maven Project will show errors after creation in this scenario.**

1. Right click on project -> JAVA EE Tools -> Generate Deployment Descriptor Stub  
more errors - look in problems window and if java version issues, follow step 2
2. Right click on project -> Properties ->
  - a. Java Compiler -> uncheck 'use compliance from 1.5' -> set compliance level to 1.8 -> apply and close
  - b. Project Facets -> set Java level to 1.8 -> apply and close
  - c. (check if a. and b. fixed the issue, if not then do this) Java Build Path -> remove JRE 5 library and Add Library -> JRE -> default 1.8 -> apply and close

**Final fix if above step 2 doesn't help,**

3. Right click on project -> maven -> update project -> force update -> repeat above steps a. and b.

**Also, if creating SERVLET and JSP files, errors will be shown**

4. Right click on project -> properties -> targeted runtimes -> check apache tomcat server -> apply and close
5. **DO:** keep .properties files in src/main/resources
6. **DO:** add JUnit dependency (if have created test cases) and Database dependency (if database is used) in pom.xml file

**To build this above project using maven build in eclipse, add below 2 to pom.xml**

(reason : maven-compiler-plugin by default will compile your project using Java 1.5 which is where m2e get's its information.

That's why you have to explicitly declare the maven-compiler-plugin in your project with something other than 1.5. Your effective pom.xml will implicitly use the default set in the maven-compiler-plugin pom.xml)

---

```
<!-- https://mvnrepository.com/artifact/javax.servlet/javax.servlet-api -->
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>javax.servlet-api</artifactId>
  <version>3.0.1</version>
  <scope>provided</scope>
</dependency>

<!-- https://maven.apache.org/plugins/maven-compiler-plugin/examples/set-compiler-source-and-target.html -->
<properties>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
</properties>
```

---

What is 'scope: provided' in servlet dependency?

- [https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html#Dependency\\_Scope](https://maven.apache.org/guides/introduction/introduction-to-dependency-mechanism.html#Dependency_Scope)

### Running a project using Maven Build in eclipse

Right click on project -> Run As -> maven build(select 1st maven build option you see) -> Goals: clean compile test package

If no options of 'run as' on project, right click on pom.xml -> Run As -> maven build(select 1st maven build option you see) -> Goals: clean compile test package

An error may occur now if doing maven build for first time in eclipse: JRE path its taking and not JDK one

**FIX:** Window -> preferences -> select java(from left side) -> Installed JREs(left side) -> Add -> Standard VM -> Next -> set JDK path (just JDK folder) to JRE home by clicking on Directory