

The image shows the Spring Initializr web interface. On the left, there are navigation icons. The main area is divided into sections: Project, Language, Spring Boot, Project Metadata, and Dependencies. The Project section has radio buttons for Gradle - Groovy, Gradle - Kotlin, and Maven (selected). The Language section has radio buttons for Java (selected), Kotlin, and Groovy. The Spring Boot section has radio buttons for 4.0.0 (SNAPSHOT), 3.5.5 (SNAPSHOT), 3.5.4, and 3.4.9 (SNAPSHOT), with 3.4.8 selected. The Project Metadata section has input fields for Group (com.ar), Artifact (logistics-models), Name (logistics-models), Description (Model library for logistic management), and Package name (com.ar.logistics-models). The Packaging section has radio buttons for Jar (selected) and War. The Java version section has radio buttons for 24 (selected), 21, and 17. The Dependencies section has a button 'ADD DEPENDENCIES... CTRL + B' and a section for Lombok (DEVELOPER TOOLS) with a description: 'Java annotation library which helps to reduce boilerplate code.' At the bottom, there are buttons 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and a button with three dots.

1. Delete `LogisticsModelsApplication.java` which is the main application from where spring boot starts include `@SpringBootApplication`
2. Delete `application.properties`
Because this is not a spring boot app but a java module

Create a Dto package and inside that your DTO/Models

```
package com.ar.logistics_models.dto;
```

```
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
@Data
@AllArgsConstructor
@NoArgsConstructor
public class TestDto {
    private String key;
    private String value;
}
```

```
Run `mvn clean install`
```

```
<img alt = "" src= "Media/Images/Creating-Model-Library-Spring-Boot-2.png" />
```

Build will fail, but jar will be generated in target and can be used in other project.

To fix this you must remove `build` from pom.xml of logistics-models

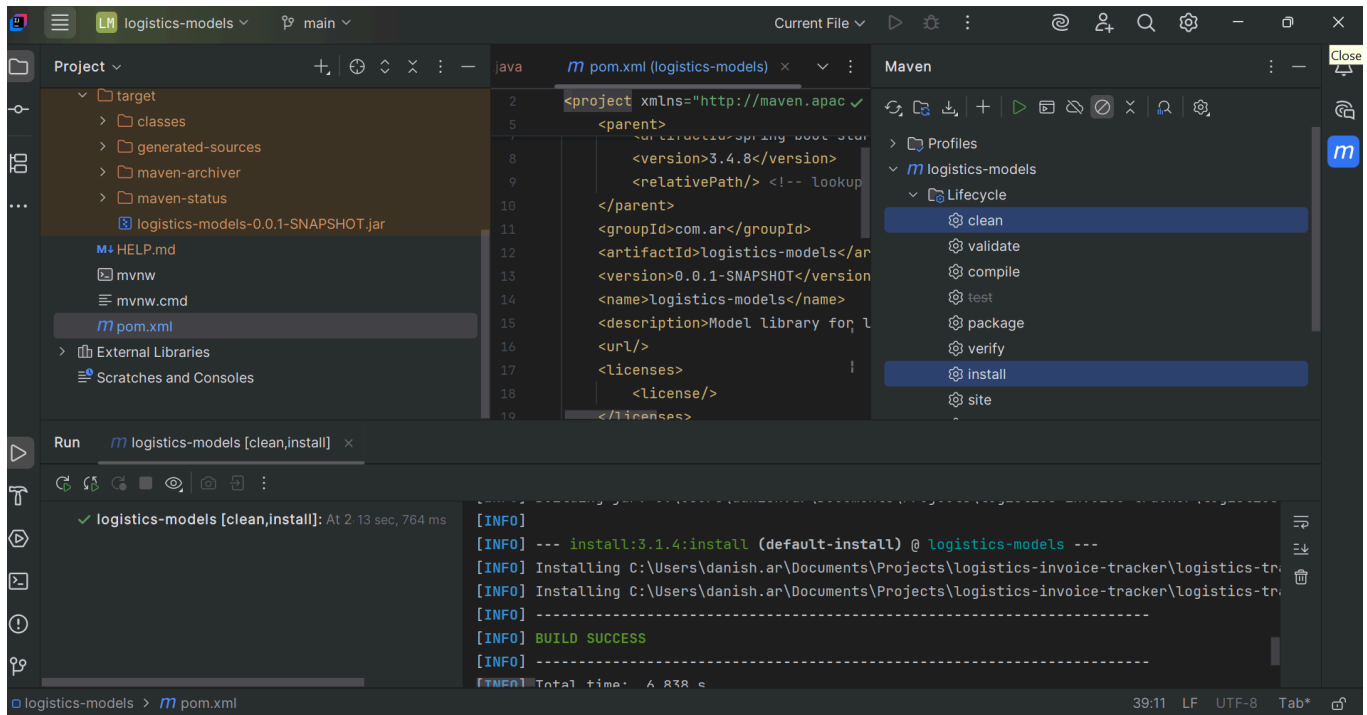
```
```xml
 <build>
 <plugins>
 <plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-compiler-plugin</artifactId>
 <configuration>
 <annotationProcessorPaths>
 <path>
 <groupId>org.projectlombok</groupId>
 <artifactId>lombok</artifactId>
 </path>
 </annotationProcessorPaths>
 </configuration>
 </plugin>
 <plugin>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-maven-plugin</artifactId>
 <configuration>
 <excludes>
 <exclude>
 <groupId>org.projectlombok</groupId>
 <artifactId>lombok</artifactId>
 </exclude>
 </excludes>
 </configuration>
 </plugin>
 </plugins>
 </build>
```

```
<dependency>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter</artifactId>
</dependency>
```

```
<dependency>
 <groupId>org.springframework.boot</groupId>
```

```
<artifactId>spring-boot-starter-test</artifactId>
<scope>test</scope>
</dependency>
```

Delete the test folder, keep if you want unit testing for models usually not required.

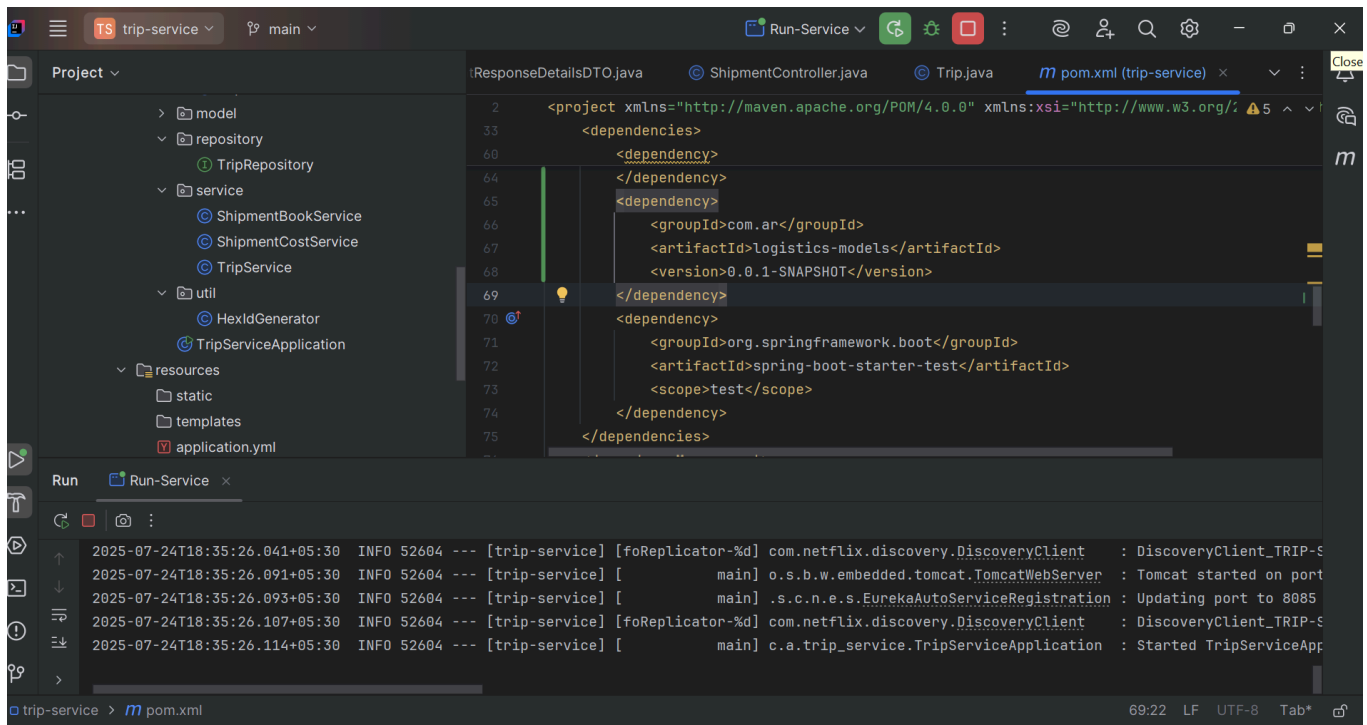


Copy the groupId, artifactId, version from pom.xml

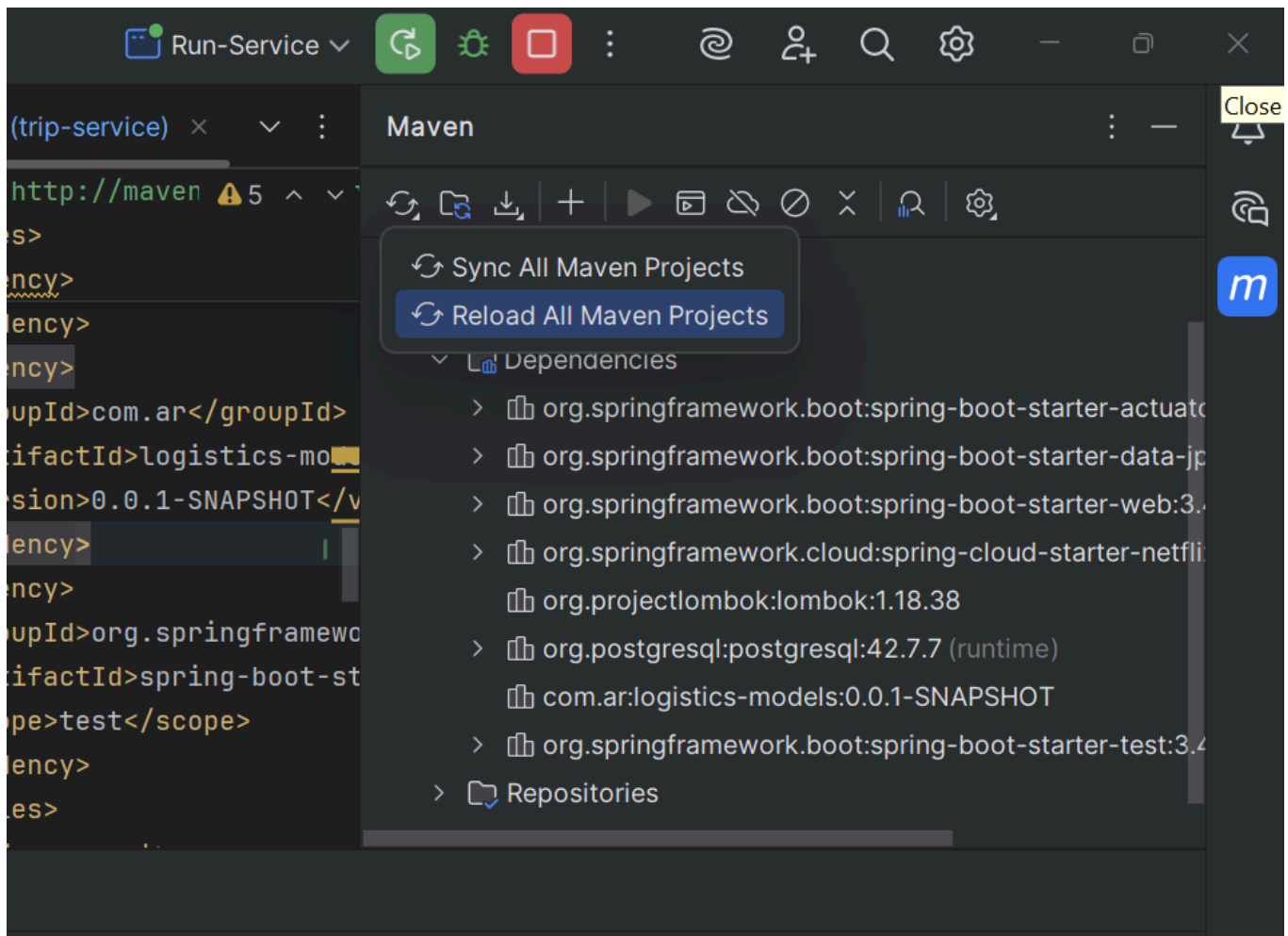
```
<groupId>com.ar</groupId>
<artifactId>logistics-models</artifactId>
<version>0.0.1-SNAPSHOT</version>
```

## Using it in trip-service

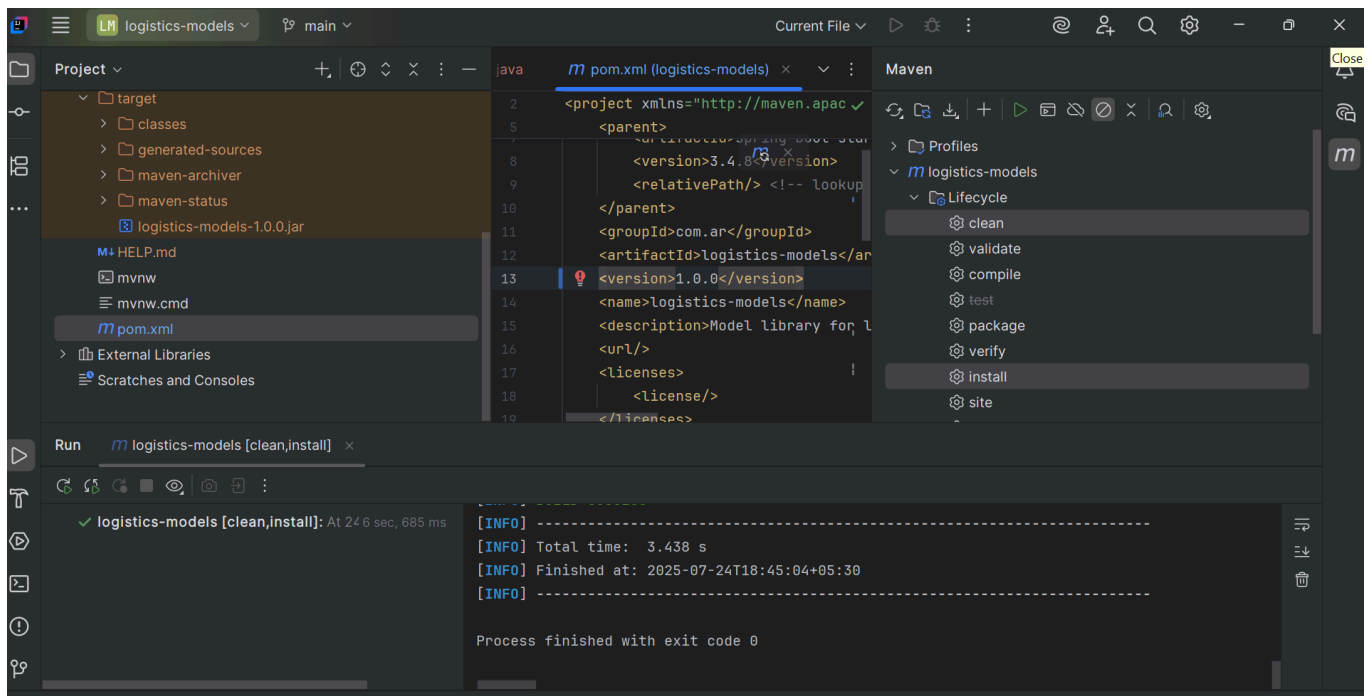
Add in the dependency



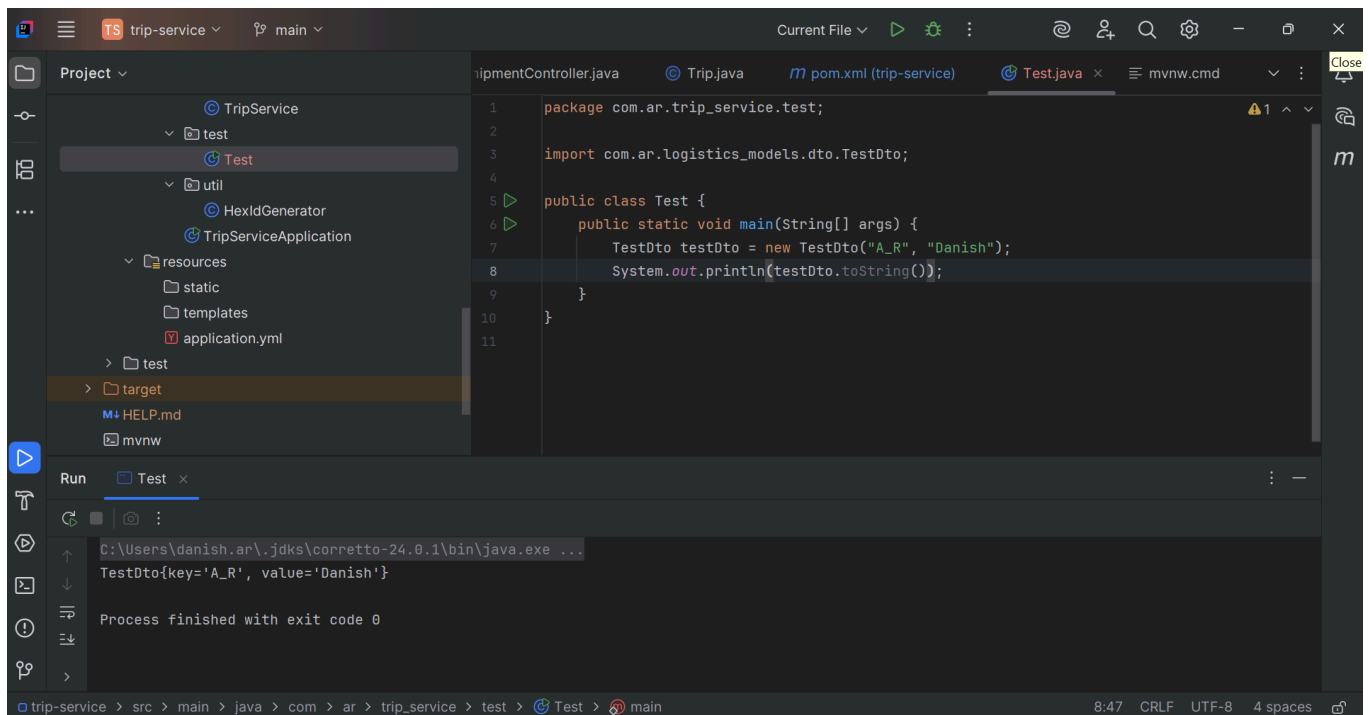
Reload maven projects:



When updating the models it is recommended to change the version

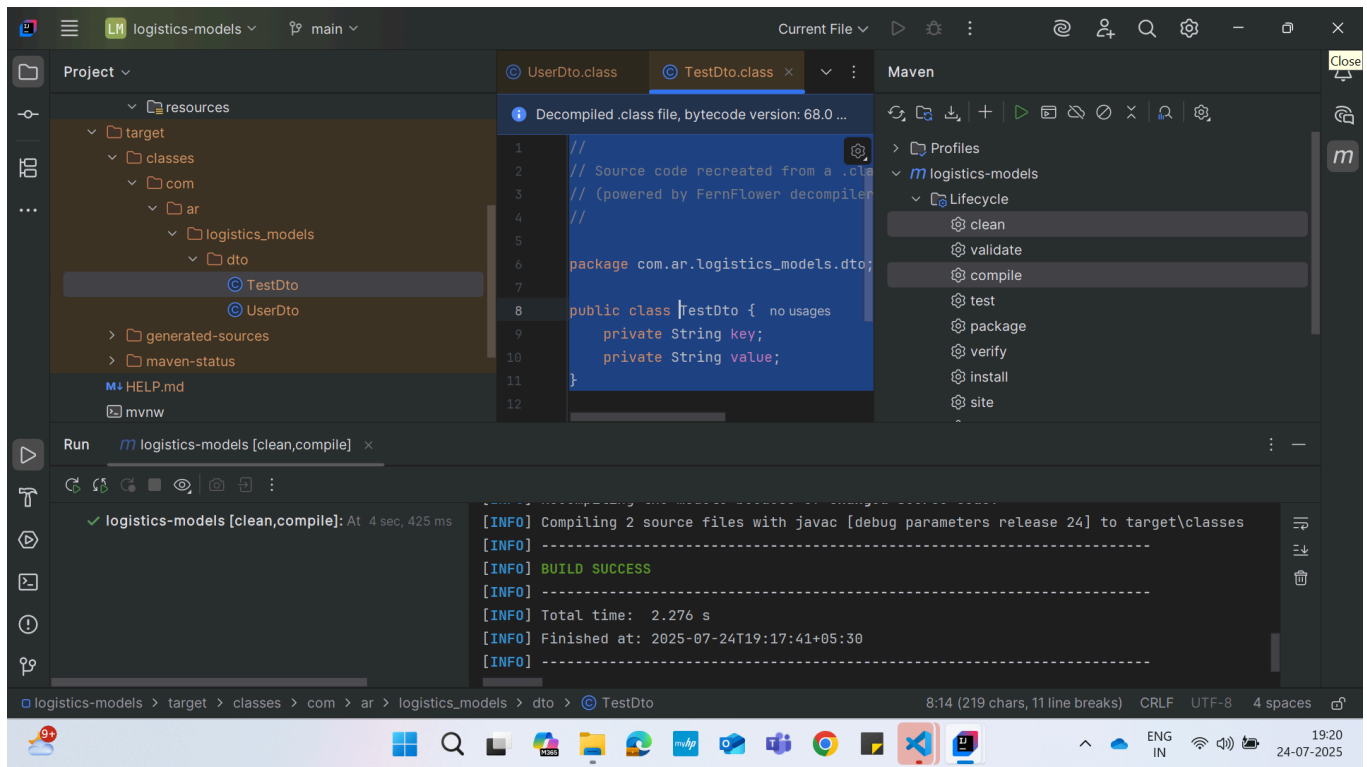


Using DTO for testing:

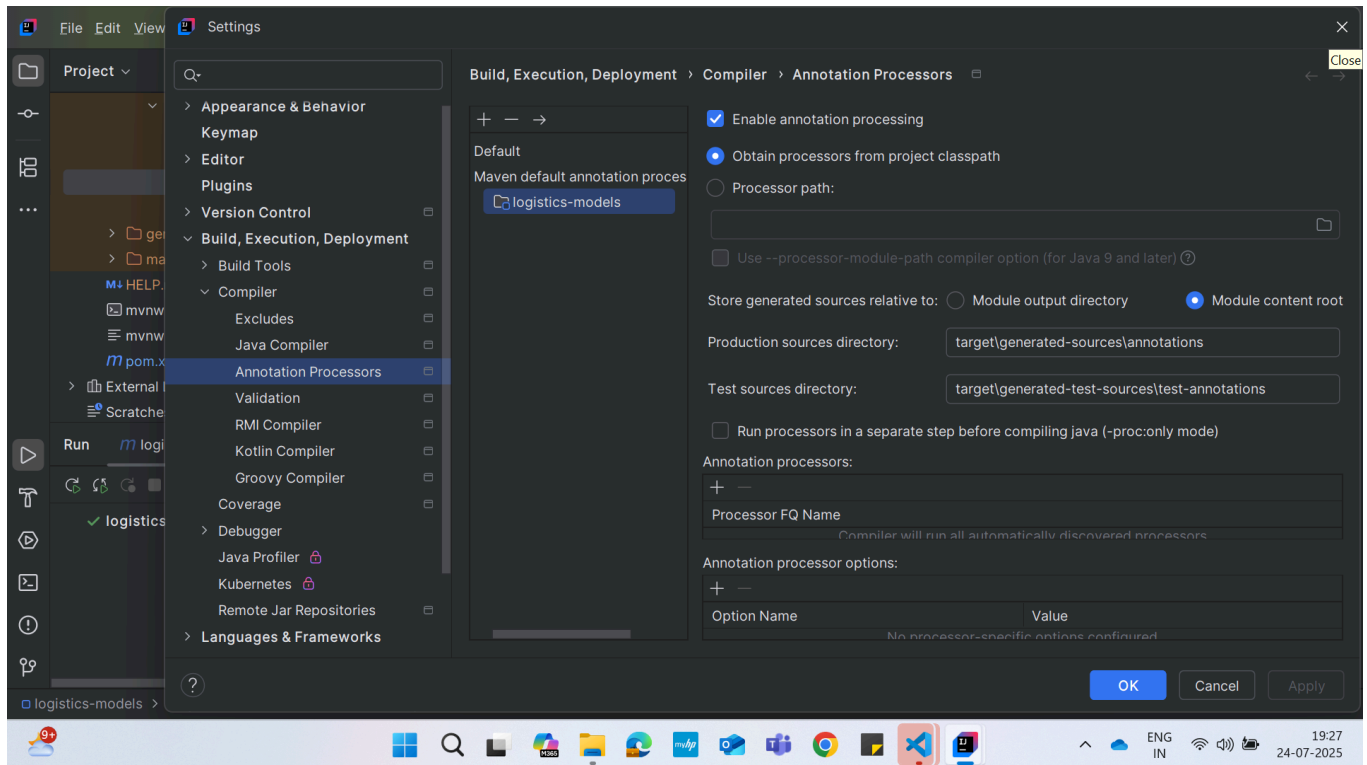


Using lombok it is not working

mvn clean compile checked the class only contains fields which is the issue



This is the issue:



Enable annotation processing was unchecked. Now checked let's recompile and verify the .class file after generation in target directory.

Still issue exists.

Actually instead of deleting whole build plugins we only need to delete: Spring boot maven plugin and keep compiler one.

Final `pom.xml` will look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <parent>
 <groupId>org.springframework.boot</groupId>
 <artifactId>spring-boot-starter-parent</artifactId>
 <version>3.4.8</version>
 <relativePath/> <!-- lookup parent from repository -->
 </parent>
 <groupId>com.ar</groupId>
 <artifactId>logistics-models</artifactId>
 <version>1.0.0</version>
 <name>logistics-models</name>
 <description>Model library for logistic management</description>
 <url/>
 <licenses>
 <license/>
 </licenses>
 <developers>
 <developer/>
 </developers>
 <scm>
 <connection/>
 <developerConnection/>
 <tag/>
 <url/>
 </scm>
 <properties>
 <java.version>24</java.version>
 </properties>
 <dependencies>
 <dependency>
 <groupId>org.projectlombok</groupId>
 <artifactId>lombok</artifactId>
 <scope>provided</scope>
 </dependency>
 </dependencies>
```

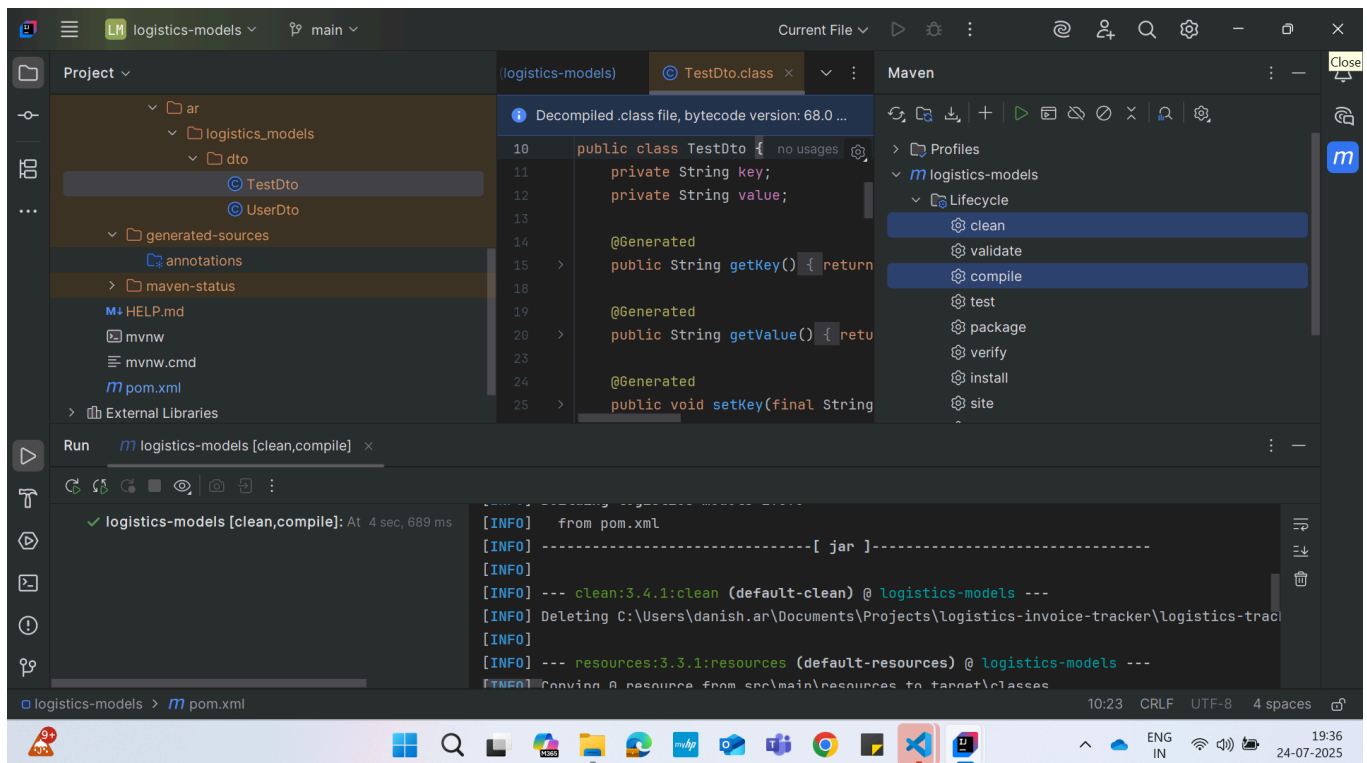
```

<build>
 <plugins>
 <plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-compiler-plugin</artifactId>
 <configuration>
 <annotationProcessorPaths>
 <path>
 <groupId>org.projectlombok</groupId>
 <artifactId>lombok</artifactId>
 </path>
 </annotationProcessorPaths>
 </configuration>
 </plugin>
 </plugins>
</build>

</project>

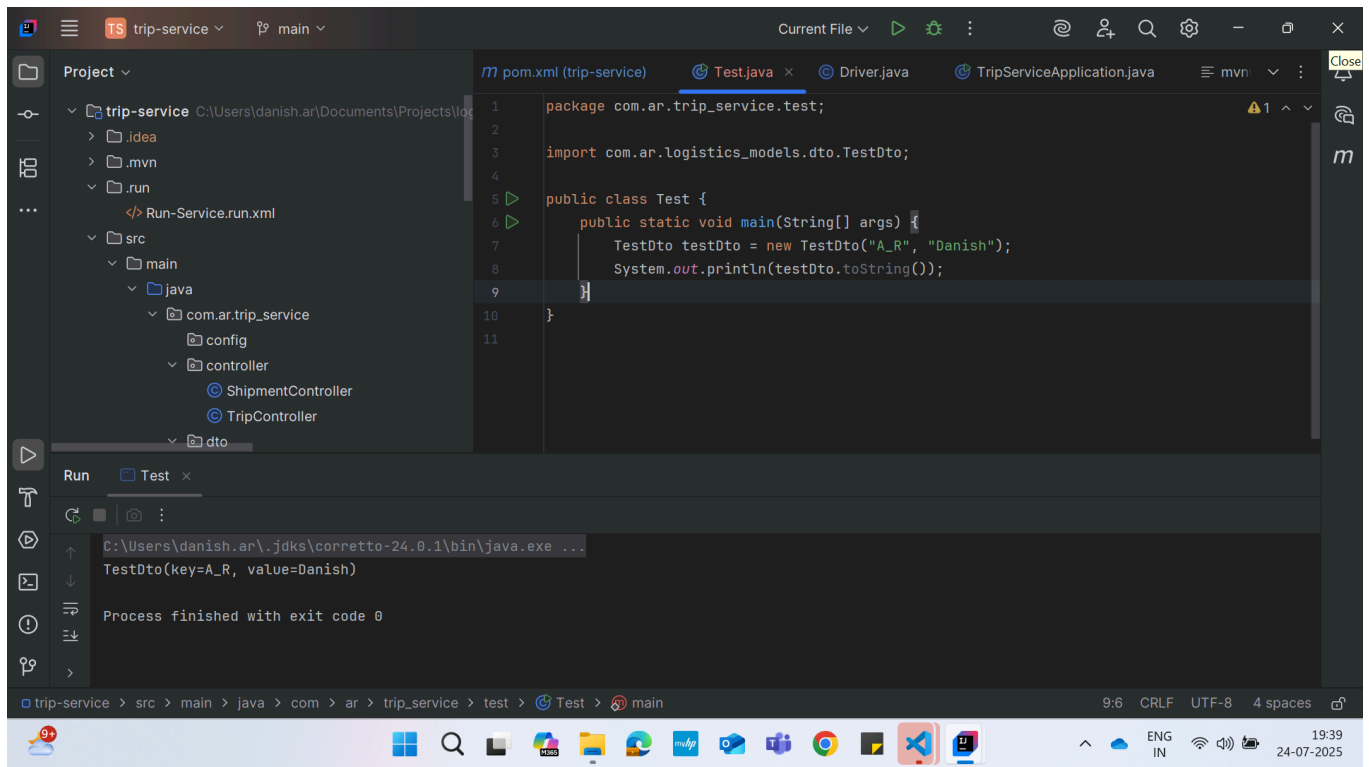
```

Now we can see getter and setter generated:



Now it also got fixed in trip-service, using **lombok** only





Happy Coding

A R

---