

Handson7: Setting Up JUnit in a Java Project

Steps we need to understand before setting up everything:

First, we need to set up JUnit in your Java project to start writing unit tests.

To understand the process of setting up JUnit in a Java project and writing unit tests using Maven and IntelliJ IDEA.

Tools & Environment Used:

- Java JDK 17
- IntelliJ IDEA Community Edition
- Maven builds tool
- JUnit 4.13.2

• *Operating System: Windows 10*

Step 1: Create a Maven Project in IntelliJ

- Open IntelliJ → New Project → Select Maven
- Choose Archetype: `maven-archetype-quickstart`
- Set GroupId and ArtifactId (e.g., `com.xyz / p1`)
- Project structure was auto-generated with `src/main/java` and `src/test/java`

Folder structure:

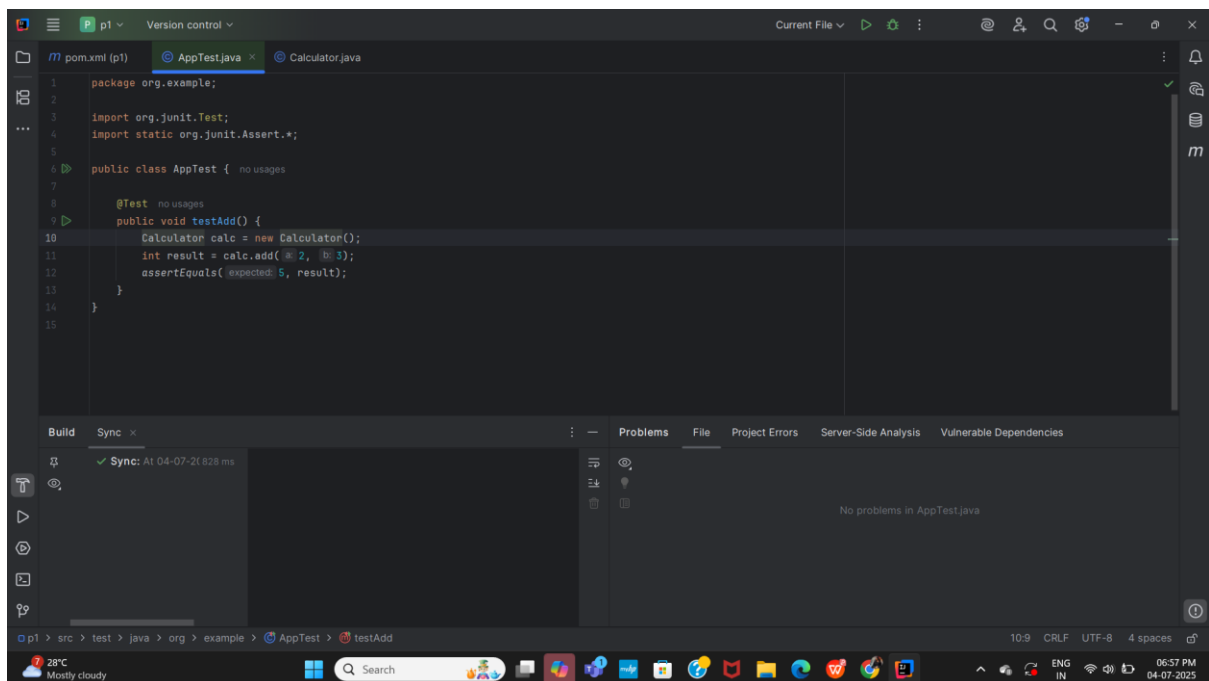
p1/

```
|— src/
|   |— main/java/org/example/Calculator.java
|   |— test/java/org/example/AppTest.java
|— pom.xml
```

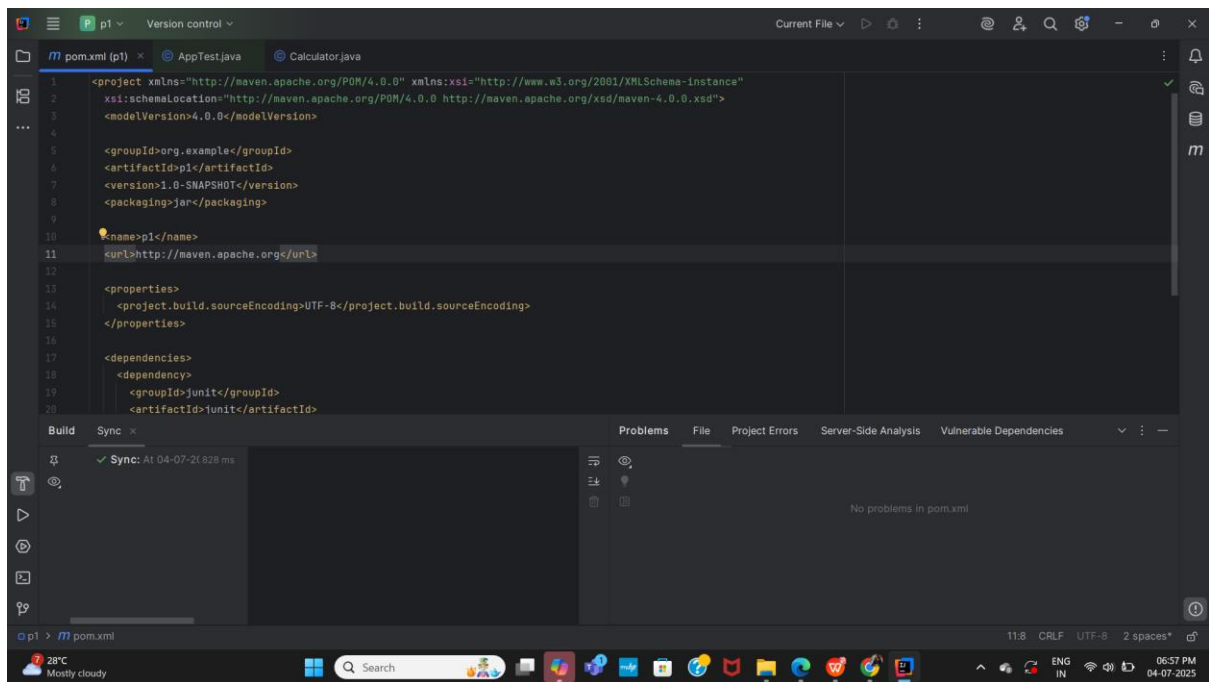
key takeaways:

- Set up a **Java project using Maven** in an IDE like IntelliJ IDEA.
- Understand the importance of **dependency management** using Maven (`pom.xml`).
- Successfully add and configure the **JUnit testing framework** to the project.
- Write a **unit test** to validate a simple method using `@Test` and `assertEquals`.
- Solve common beginner issues like:
 - Class not matching file name
 - Missing dependencies
 - Maven not syncing properly
- Realize the value of testing code in **smaller components (unit testing)** to ensure quality.

Output:



1.Green icon of successful build



Pom file

