

## TD 1 - Java et Spring Boot

### Version du langage JAVA

Suivre la version d'un langage (son évolution) :

Pour le JDK 21 LTS (ou 24), allez sur le site d'Oracle pour voir les nouveautés apportées par cette version :

<https://docs.oracle.com/en/java/javase/21/language/java-language-changes-release.html#GUID-6459681C-6881-45D8-B0DB-395D1BD6DB9B>

### Utiliser IntelliJ Ultimate :

The screenshot shows the 'New Project' dialog in IntelliJ IDEA. On the left, under 'Generators', 'Spring Boot' is selected. The main area contains the following configuration:

- Server URL: start.spring.io
- Name: demo1
- Location: ~\IdeaProjects (Project will be created in: ~\IdeaProjects\demo1)
- ☐ Create Git repository
- Language: Java (selected), Kotlin, Groovy
- Type: Gradle - Groovy (selected), Gradle - Kotlin, Maven
- Group: com.example
- Artifact: demo1
- Package name: com.example.demo
- JDK: 21 Oracle OpenJDK 21.0.7
- Java: 21 (selected in dropdown)
- Packaging: Jar (selected), War

At the bottom, there are 'Next' and 'Cancel' buttons.

## Fonctionnement Lombok et Spring :

Installer demo-lombok.zip dans IdeaProjects, l'ouvrir avec IntelliJ.

Comprendre les classes POJO : Plain Old Java Object

<https://medium.com/@Bharat2044/what-is-pojo-class-in-java-1325eb7f3a6c>

Quels commentaires pouvez-vous faire sur POJO ?

## Travail à faire :

Faire l'exemple suivant en Spring Boot avec Maven avec la mise en oeuvre de lombok :

<https://springjava.com/spring-boot/use-lombok-in-spring-boot/>

- avec la méthode classique (POJO)
- avec lombok et l'annotation @Data
- avec lombok et l'annotation @Build (et ajout d'annotations autres si nécessaire)

Pour l'annotation @Build, vous pouvez rechercher de la documentation en ligne ou vous inspirer de <https://www.baeldung.com/lombok-builder> .

Quelles sont les différences ? Intérêt de lombok ? Inconvénients ?

## Travail à faire :

Suivre le tutoriel ci-dessous, lancez l'application et testez les endpoints avec POSTMAN :

<https://www.geeksforgeeks.org/springboot/spring-boot-how-to-access-database-using-spring-data-jpa/>

## Créez le fichier exécutable .jar et exécutez-le :

Au préalable, vérifier que la version Java de votre IDE (au niveau du compilateur) et celle de votre environnement Windows correspondent (version Java 21 LTS).

Vérifier votre variable d'environnement pour Java, ici JAVA\_HOME, et votre version de Java pour Windows : java -version :

```
C:\Users\caspar5>echo %JAVA_HOME%
C:\Program Files\Java\jdk-21

C:\Users\caspar5>java --version
java 21.0.7 2025-04-15 LTS
Java(TM) SE Runtime Environment (build 21.0.7+8-LTS-245)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.7+8-LTS-245, mixed mode, sharing)
```

Lancez la commande Maven suivante depuis la racine de votre projet :

mvn clean package -DskipTests

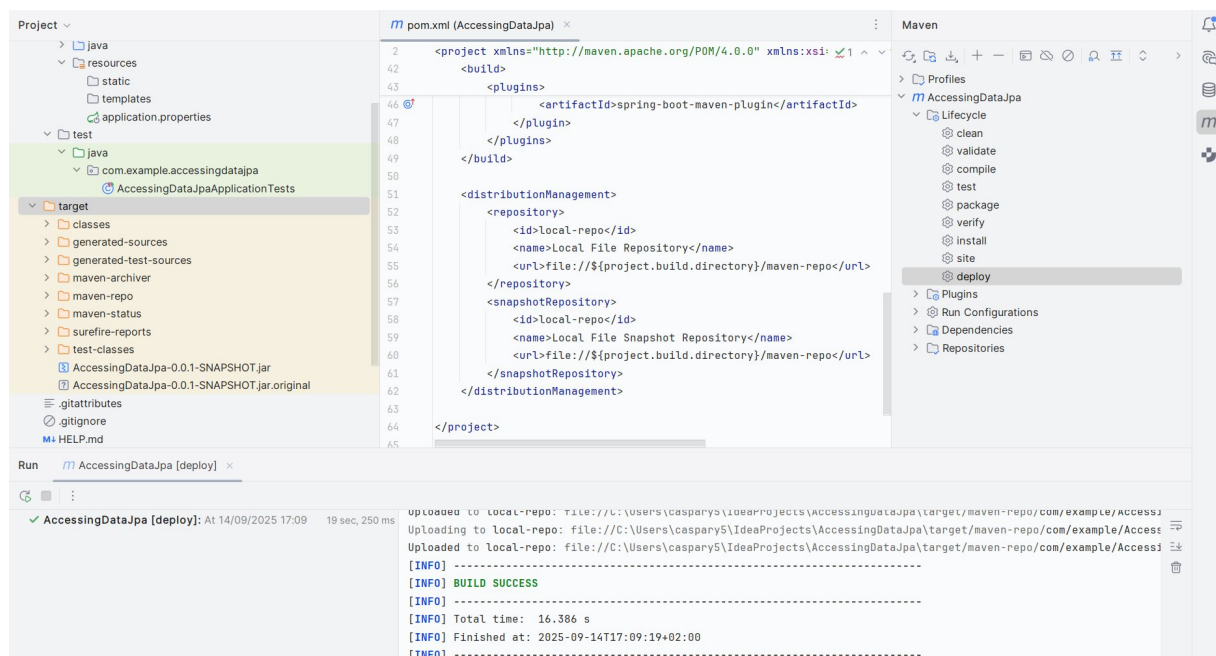
Soit (à faire sur une seule ligne) :

```
C:\Users\caspar5\IdeaProjects\AccessingDataJpa>  
mvn clean package -DskipTests
```

Vous pouvez maintenant exécuter votre application avec la commande (à faire sur la même ligne et à partir du répertoire target) :

```
C:\Users\caspar5\IdeaProjects\AccessingDataJpa\target>  
java -jar AccessingDataJpa-0.0.1-SNAPSHOT.jar
```

Autre possibilité directement depuis IntelliJ : maven -> deploy



Mais au préalable dans pom.xml, à la fin juste avant </project>, installez :

```
<distributionManagement>  
  <repository>  
    <id>local-repo</id>  
    <name>Local File Repository</name>  
    <url>file://${project.build.directory}/maven-repo</url>  
  </repository>  
  <snapshotRepository>  
    <id>local-repo</id>  
    <name>Local File Snapshot Repository</name>  
    <url>file://${project.build.directory}/maven-repo</url>  
  </snapshotRepository>  
</distributionManagement>
```