



# Architecture and framework web

Création d'une application web CRUD  
2025-2026

MARBOIS Bryce  
SANOUILLER-TOURNE Timothé

Capgemini 



Bryce MARBOIS

2016 – Licence Informatique  
2018 – Master INIS  
2018 – Software Engineer  
2022 – Techlead



Timothé SANOUILLER--TOURNE

2017 – Diplôme BTS SIO  
2020 – Diplôme Polytech Tours  
2021 – Arrivée chez Capgemini  
2023 – Techlead



# Objectif du module

## Notions

- Comprendre les notions basic de Spring et d'Angular
- Mettre en place une interconnexion entre le front, le back et la base de données du projet
- Mise en pratique de vos compétences grâce à un projet en autonomie partielle.



# Objectif du module

- DEMO



# Mise en place d'une API Java Spring

Plan

TP 1 : Mise en place d'un projet Spring, des objets métiers et d'une base de données

TP2 : Mise en place de repositories et de services manipulant les objets métiers

TP3 : Mise en place de la couche d'exposition



# Mise en place d'une API Java Spring

Plan

TP 1 : Mise en place d'un projet Spring, des objets métiers et d'une base de données

TP2 : Mise en place de repositories et de services manipulant les objets métiers

TP3 : Mise en place de la couche d'exposition



# Scripts de création des bases

Les scripts de création et de peuplement des bases de données se trouvent sur git :

- <https://gitlab.com/capgemini-polytech/databases>

The screenshot shows a GitLab repository interface for the project 'databases'. The repository has a single commit titled 'Merge branch 'master' into 'main'' by 'Timothé Sacoullier' (Tourne autorisé il y a 3 jours). The commit hash is f2c37665. The repository contains the following files:

Nom	Dernière validation	Dernière mise à jour
aire de jeux	Update 14 files	Il y a 3 jours
boite livres	Update 14 files	Il y a 3 jours
composteur	Update 14 files	Il y a 3 jours
parking velo	Update 14 files	Il y a 3 jours
pmr	Update 14 files	Il y a 3 jours
terrain de petanque	Update 14 files	Il y a 3 jours
velociste	Update 14 files	Il y a 3 jours
README.md	Update 14 files	Il y a 3 jours

The 'README.md' file contains the following text:

```
databases

Ce projet contient l'ensemble des jeux de données à disposition des étudiants.
```



# Création de base de données

Pour créer la base de données :

- Ouvrir un terminal à l'endroit où est installer MariaDB
- Connectez-vous à MariaDB : mariadb.exe -u root
- Créer une base de données : CREATE DATABASE exemple;

```
C:\Program Files\MariaDB 11.5\bin
./mariadb.exe -u root
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 68
Server version: 11.5.2-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

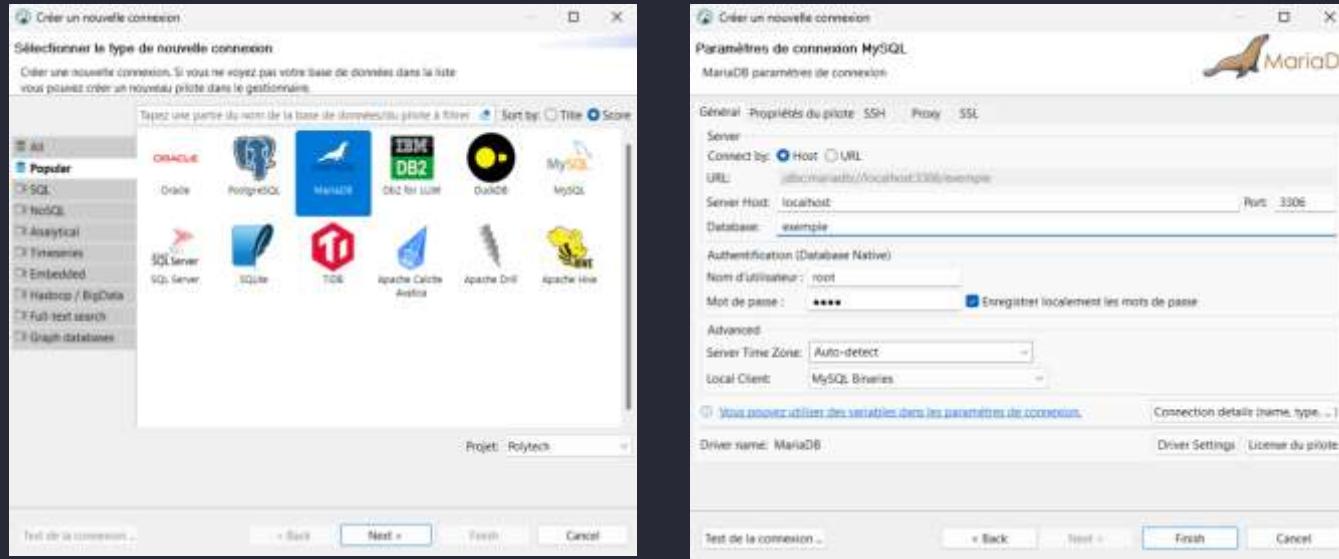
MariaDB [(none)]> CREATE DATABASE exemple;
Query OK, 1 row affected (0.004 sec)

MariaDB [(none)]>
```

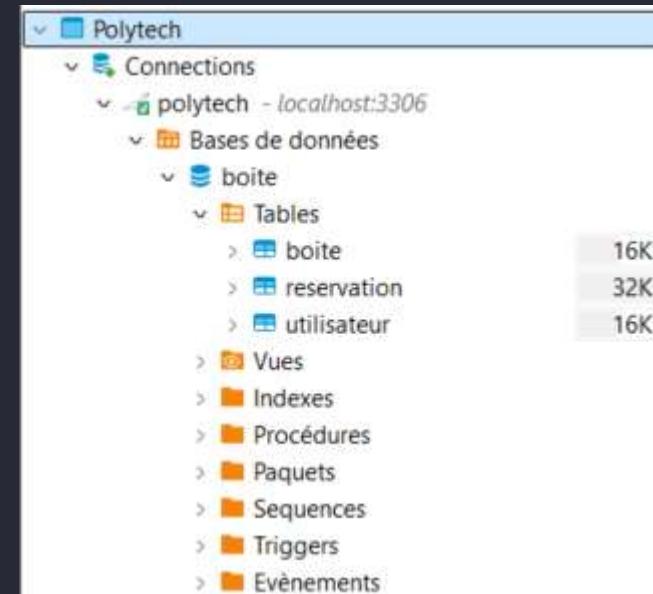


# Création de la base de données

Créer une connexion sur Dbeaver vers la base de données créée via Mariadb :



Jouer le script de création de la structure et d'insertion des données





# Initialisation du projet

<https://start.spring.io/#!type=maven-project&language=java&platformVersion=3.4.10&packaging=jar&jvmVersion=21&groupId=com.polytech&artifactId=polytech&name=polytech&description=Projet%20Polytech%202025&packageName=com.polytech.polytech&dependencies=web,lombok,data-jpa,mariadb>

The screenshot shows the Spring Initializr web interface with the following configuration:

- Project:** Maven (selected)
- Language:** Java (selected)
- Spring Boot:** 3.4.10 (selected)
- Project Metadata:**
  - Group: com.polytech
  - Artifact: polytech
  - Name: polytech
  - Description: Projet Polytech 2025
  - Package name: com.polytech.polytech
  - Packaging: Jar (selected)
  - Java: 21 (selected)
- Dependencies:**
  - Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
  - Lombok** (DEVELOPER TOOLS): Java annotation library which helps to reduce boilerplate code.
  - Spring Data JPA** (SQL): Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.
  - MariaDB Driver** (SQL): MariaDB JDBC and R2DBC driver.
- ADD DEPENDENCIES... CTRL + B** button



# Configuration projet

Dans le fichier `src/main/resources/application.yml` il faut configurer la connexion à la base de données:

- `spring.datasource.url` : Lien vers la base de données
- `spring.datasource.username`: nom de l'utilisateur de la base de données
- `spring.datasource.password` : mdp de l'utilisateur de la base de données
- `spring.datasource.driver-class-name`: driver correspondant au type de SGBD utilisé

```
1 <dependency>
2   <groupId>org.mariadb.jdbc</groupId>
3   <artifactId>mariadb-java-client</artifactId>
4   <version>3.3.3</version>
5 </dependency>
```

```
---
spring:
  application:
    name: opendata-crud
  datasource:
    url: jdbc:mariadb://localhost:3306/polytech
    username: root
    password: mypass
    driver-class-name: org.mariadb.jdbc.Driver
```



# Entités et lombok

1. Créez un package src/main/ src/main/votre-projet/entity
2. Dans ce package, créez une classe java par table de la base de données, il faudra que chaque champ de la table soit représenté par un attribut
3. Vous pourrez utiliser Lombok pour générer des méthodes à l'aide d'annotation  
<https://projectlombok.org/features/>

