Architecture and framework web

Création d'une application web CRUD
Octobre 2024

MARBOIS Bryce
SANOUILLER--TOURNE Timothé
BOULLAND Romain

Capgemini

Sources des données

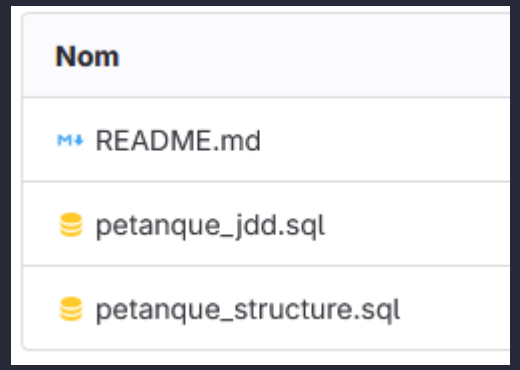
Fichiers sources provenant de Tours open data (https://data.tours-metropole.fr/pages/home/):

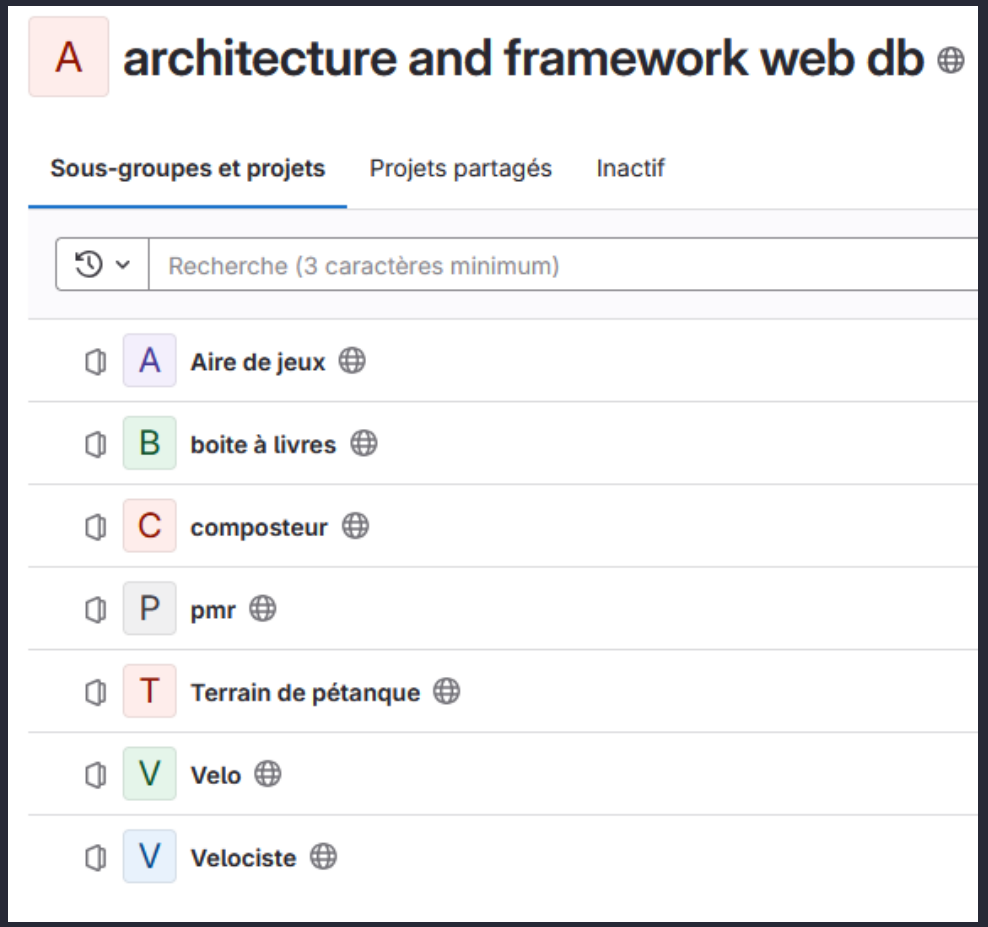
- https://data.tours-metropole.fr/explore/dataset/aire-de-jeux-tours-metropole-val-de-loire/table/
- https://data.tours-metropole.fr/explore/dataset/boites-a-livre-tours-metropole-val-de-loire/table/
- https://data.tours-metropole.fr/explore/dataset/composteurs-partages-tours-metropole-val-de-loire/table/?disjunctive.ville
- https://data.tours-metropole.fr/explore/dataset/stationnement-pmr-tmvl/table/?disjunctive.commune&disjunctive.nbre_place
- https://data.tours-metropole.fr/explore/dataset/terrain-de-petanque-tmvl/table/
- https://data.tours-metropole.fr/explore/dataset/stationnement-velo-tours-metropole/table/?disjunctive.abri
- https://data.tours-metropole.fr/explore/dataset/velocistes-tours-metropole/table/

Scripts de création des bases

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

- https://gitlab.com/architecture-and-framework-web-db/





Création de base de données

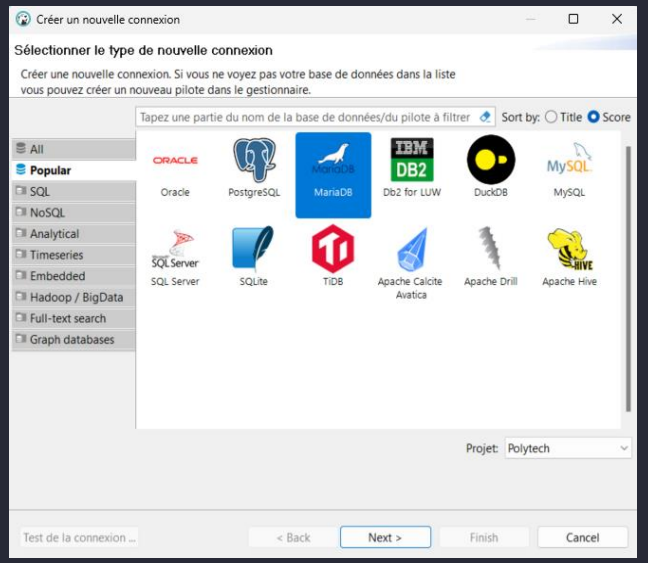
Pour créer la base de données :

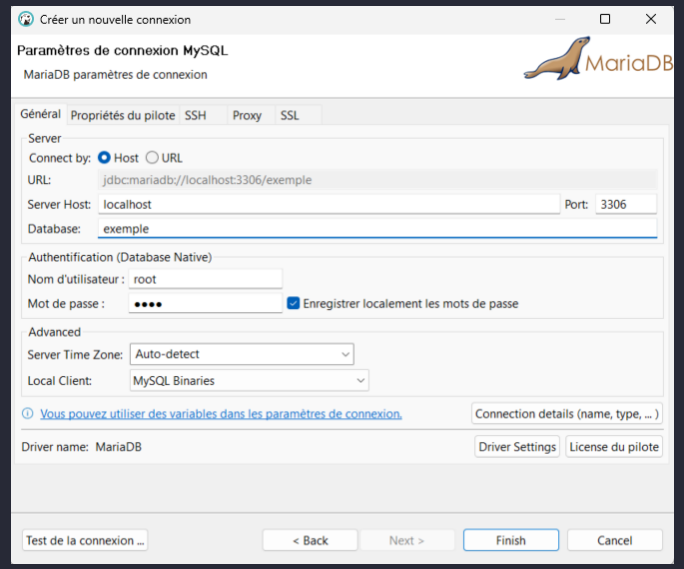
- Ouvrir un terminal à l'endroit où est installer MariaDB
- Connectez-vous à MariaDB : mariadb.exe –u root
- Créer une base de données : CREATE DATABASE exemple;

```
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)]>
```

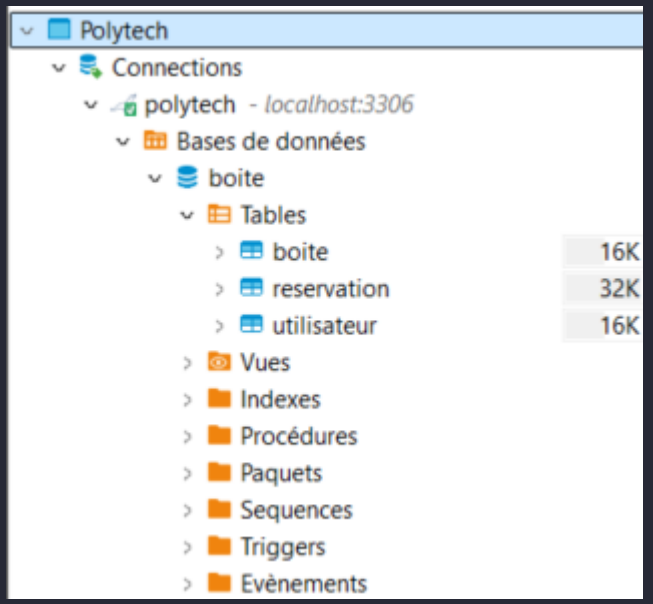
Création de la base de données

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



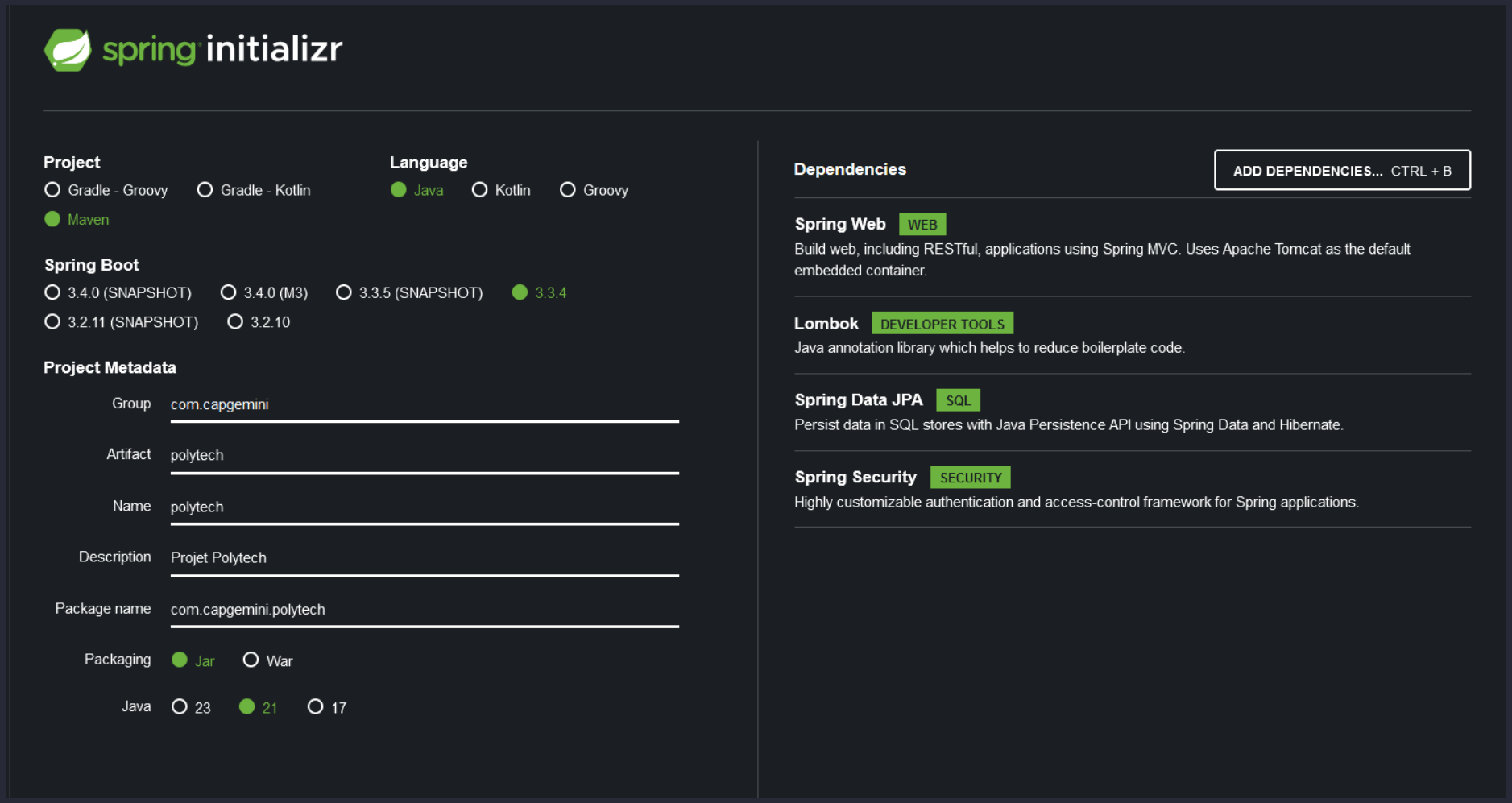


Jouer le script de création de la structure et d'insertion des données



Initialisation du projet

https://start.spring.io/#!type=maven-project&language=java&platformVersion=3.3.4&packaging=jar&jvmVersion=17&groupId=com.example&artifactId=demo&name=demo&description=Demo%20project%20for%20Spring%20Boot&packageName=com.example.demo&dependencies=web,lombok,data-jpa,security,mariadb



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

```xml
<dependency>
    <groupId>org.mariadb.jdbc</groupId>
    <artifactId>mariadb-java-client</artifactId>
    <version>3.3.3</version>
</dependency>
```

```yaml
---
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

https://projectlombok.org/features/

```java
import jakarta.persistence.*;
import lombok.*;



@Entity
@Builder
@Getter
@AllArgsConstructor(access = AccessLevel.PRIVATE)
@NoArgsConstructor
@ToString
@Table(name="tree")
public class TreeEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;

    @NonNull
    private String genre;

    @NonNull
    private String espece;

    @NonNull
    private String geoPoint;

    @NonNull
    private Integer capacity;

}

```

DTO et Mapstruct

```java

import lombok.*;


@Builder
@Getter
@Setter
@AllArgsConstructor(access = AccessLevel.PRIVATE)
@NoArgsConstructor
@ToString
@EqualsAndHashCode
public class TreeDTO {

    private Integer id;
    private String genre;
    private String espece;
    private String geoPoint;
    private Integer capacity;
}
```

```java

import com.cours.opendata_crud.dto.TreeDTO;
import com.cours.opendata_crud.entity.TreeEntity;
import org.mapstruct.Mapper;


@Mapper(componentModel = "spring")
public interface TreeMapper {

    TreeDTO toDTO(TreeEntity treeEntity);


    TreeEntity toEntity(TreeDTO treeDTO);
}

```

Configuration mapstruct

Ajouter la configuration mapstruct de dans le fichier pom.xml:

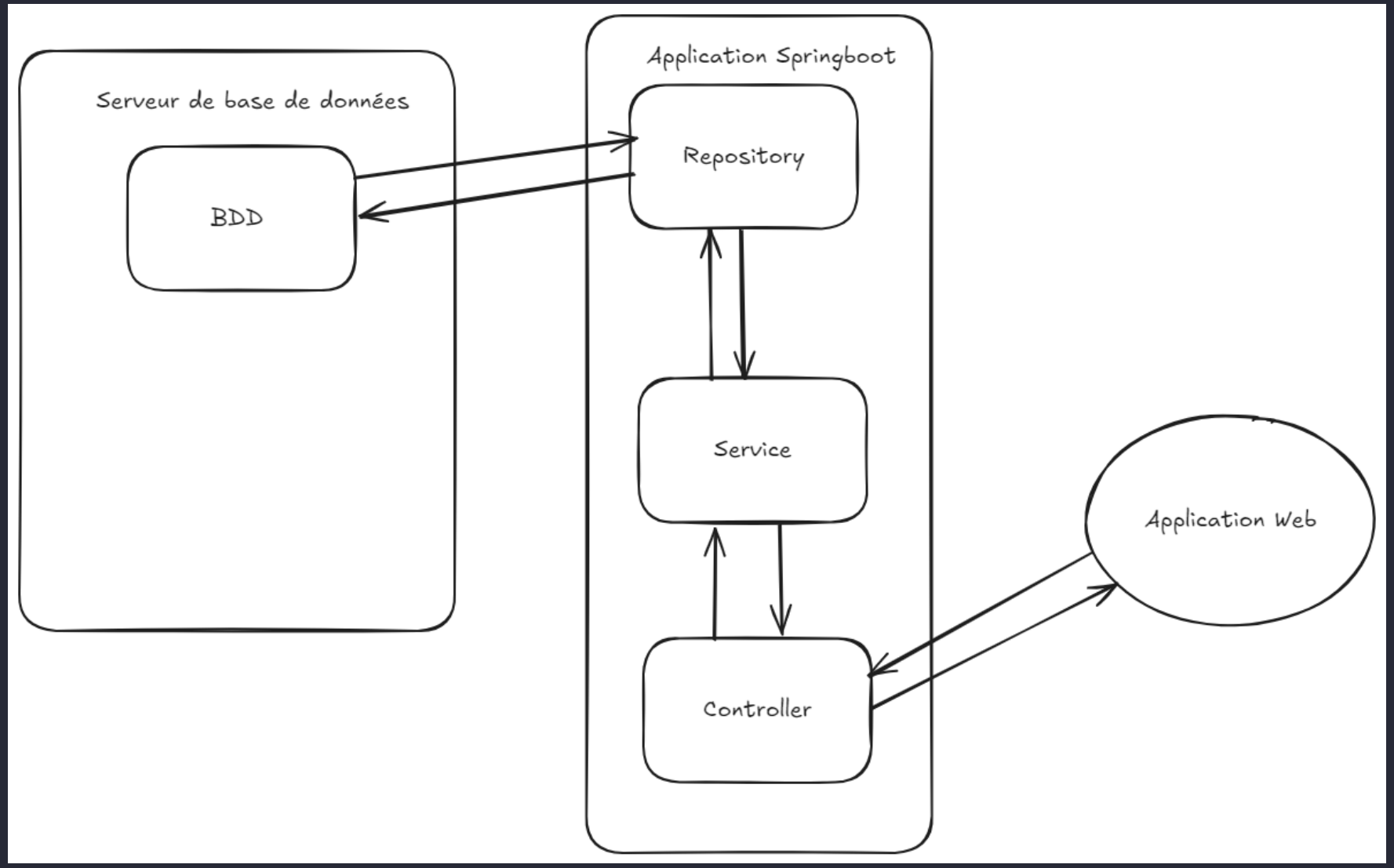
```xml
<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-compiler-plugin</artifactId>
            <version>3.11.0</version>
            <configuration>
                <annotationProcessorPaths>
                    <path>
                        <groupId>org.mapstruct</groupId>
                        <artifactId>mapstruct-processor</artifactId>
                        <version>1.5.5.Final</version>
                    </path>

                    <path>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok</artifactId>
                        <version>1.18.34</version>
                    </path>

                    <!-- This is needed when using Lombok 1.18.16 and above -->
                    <path>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok-mapstruct-binding</artifactId>
                        <version>0.2.0</version>
                    </path>
                </annotationProcessorPaths>
            </configuration>
        </plugin>
    </plugins>
</build>
```

Correction TP1

Code : https://gitlab.com/architecture-and-framework-web-db/application-crud-spring-back

Branche : tp1

Architecture du projet



JPA Repository

Créer le @Repository Spring Data correspondant à l'entité choisie

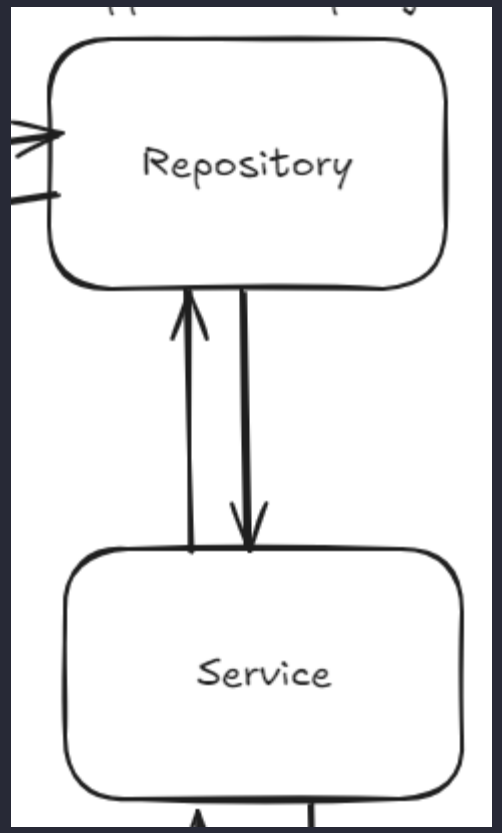
```
1
2  import com.cours.opendata_crud.entity.TreeEntity;
3  import org.springframework.data.jpa.repository.JpaRepository;
4  import org.springframework.stereotype.Repository;
5
6  @Repository
7  public interface TreeRepository extends JpaRepository<TreeEntity, Integer> {
8  }
9
```

Service

Créer un @Service Spring permettant d'effectuer les opérations CRUD sur le repository :

- Création
- Récupération unitaire
- Récupération par liste
- Mise à jour
- Suppression

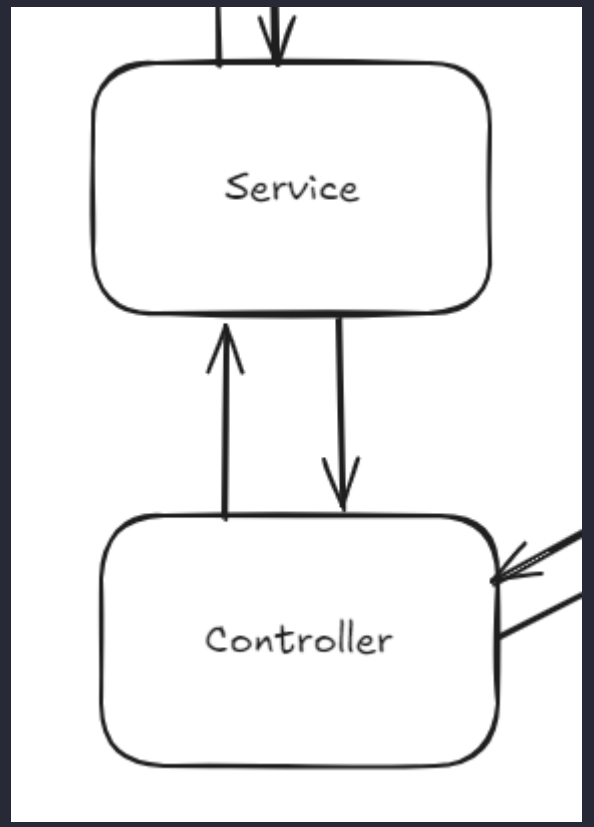
- Le service se charge d'appeler le Mappeur pour les conversions entre DTO et entité.



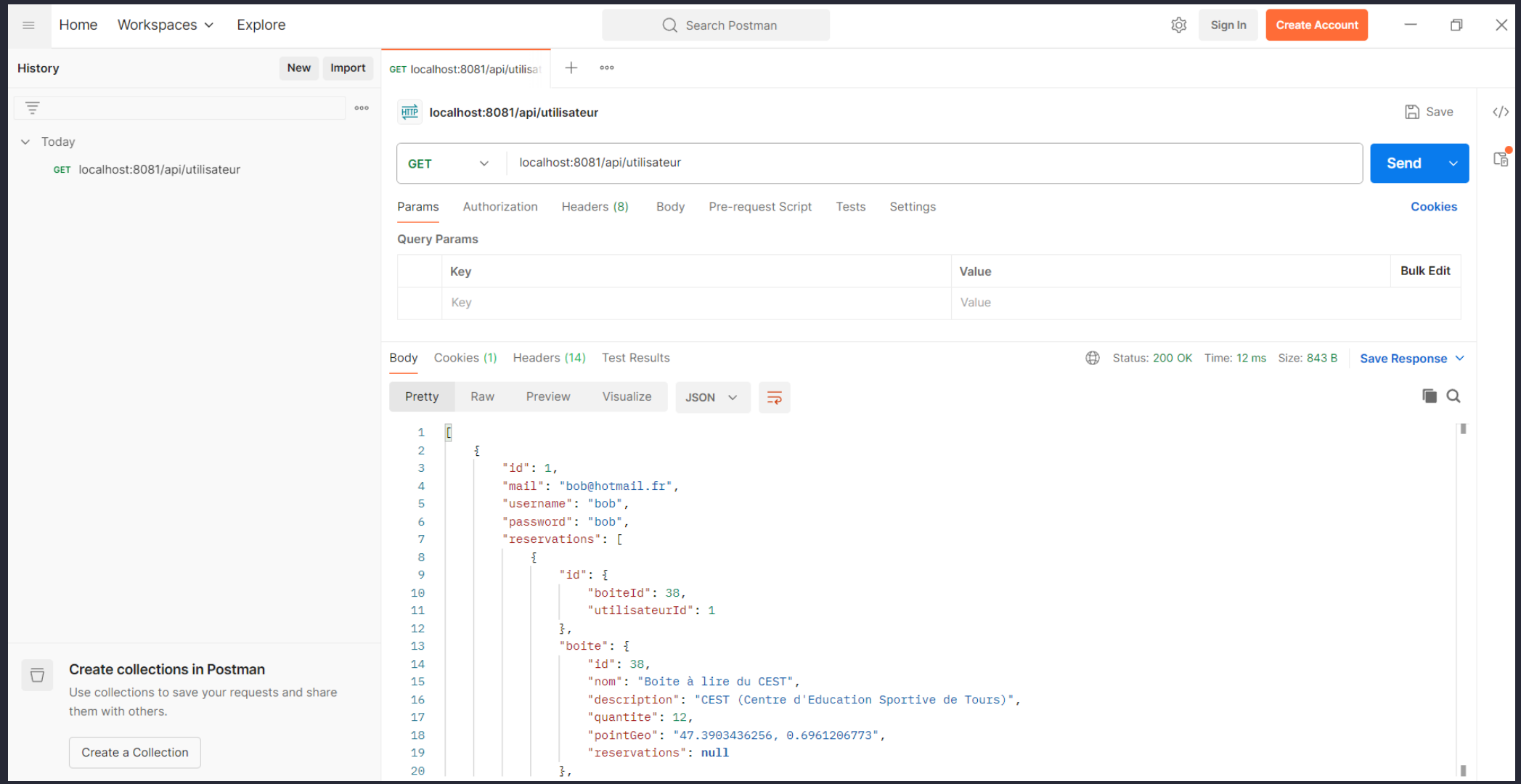
Controller

Créer un @Controller Spring permettant d'accéder aux méthodes du services

Celles-ci doivent permettre d'accéder aux opérations CRUD du Service en utilisant les actions HTTP appropriées.



Outils type bruno // postman



About Capgemini

Capgemini is a global business and technology transformation partner, helping organizations to accelerate their dual transition to a digital and sustainable world, while creating tangible impact for enterprises and society. It is a responsible and diverse group of 340,000 team members in more than 50 countries. With its strong over 55-year heritage, Capgemini is trusted by its clients to unlock the value of technology to address the entire breadth of their business needs. It delivers end-to-end services and solutions leveraging strengths from strategy and design to engineering, all fueled by its market leading capabilities in AI, cloud and data, combined with its deep industry expertise and partner ecosystem. The Group reported 2023 global revenues of €22.5 billion.

Get the future you want | www.capgemini.com

Capgemini