Spring Boot avec l’architecture hexagonale

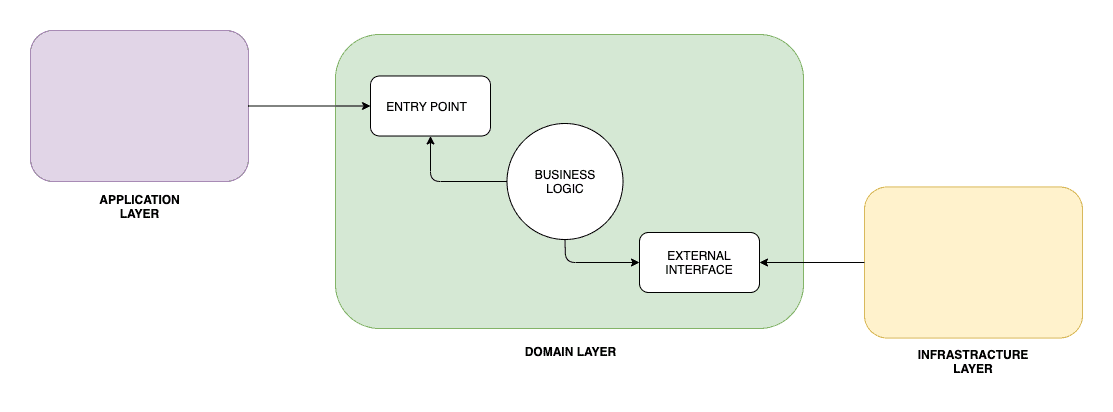
(API REST)

Pour ce tp nous allons mettre en place une application de gestion des utilisateurs (API REST – MYSQL ou MONGODB) en utilisant l’architecture hexagonale pour structurer l’application spring boot.

C’est quoi l’architecture hexagonale ?

L’architecture hexagonale est un modèle de conception d’applications logicielles autour de la logique de domaine pour l’isoler des facteurs externes.

La logique du domaine est spécifiée dans le cœur métier, que nous appellerons la partie interne, le reste étant des parties externes. L’accès à la logique du domaine depuis l’extérieur est disponible via des ports et des adaptateurs.



Les 3 trois principes :

La couche applicative (application)

Grâce à la couche applicative, le programme interagit avec l’application. Cette zone doit contenir des éléments tels que des interfaces utilisateurs, API RESTFUL et des bibliothèques de sérialisation JSON. Elle comprend tout ce qui expose à l’entrée dans notre application et orchestre l’exécution de la logique du domaine.

La couche domaine (domain)

Dans la couche domaine, nous gardons le code qui touche et implémente la logique métier. C’est le cœur de notre application. Cette couche doit être isolée à la fois de la partie application et de la partie infrastructure. De plus, elle doit également contenir des interfaces qui définissent l’API pour communiquer avec des parties externes, comme la base de données, avec lesquelles le domaine interagit.

La couche infrastructure (infrastructure)

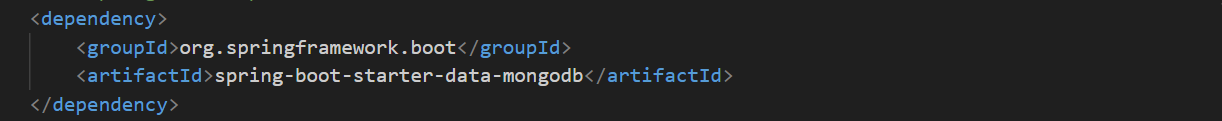
La couche d’infrastructure est la partie qui contient tout ce dont l’application a besoin pour fonctionner, comme la configuration de la base de données ou la configuration de spring. Elle implémente également les interfaces dépendantes de l’infrastructure à partir de la couche de domaine.

Mise en place de l’application

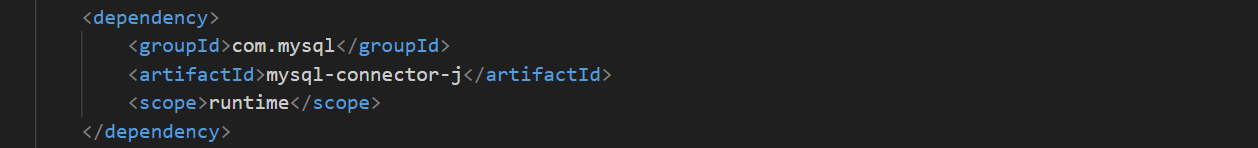
Nous allons mettre en place notre architecture hexagonale comme celle-ci :



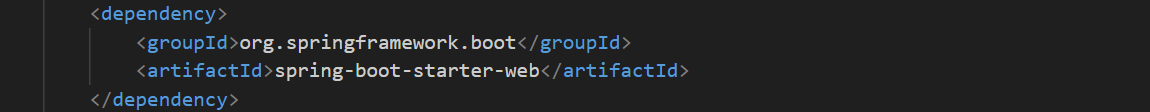
On ajoute la dépendance mongodb



On ajoute la dépendance mysql

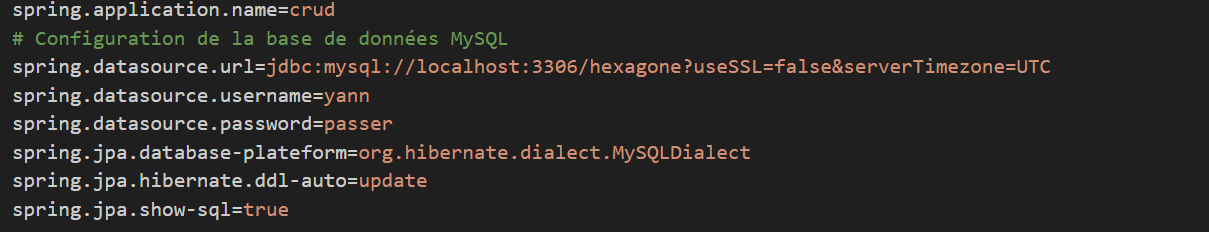


Et on ajoute la dépendance pour web

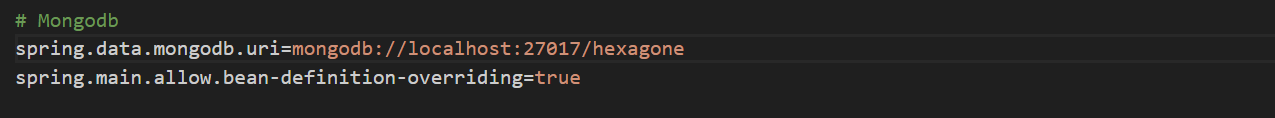


On configure notre connexion aux bases de données application.properties

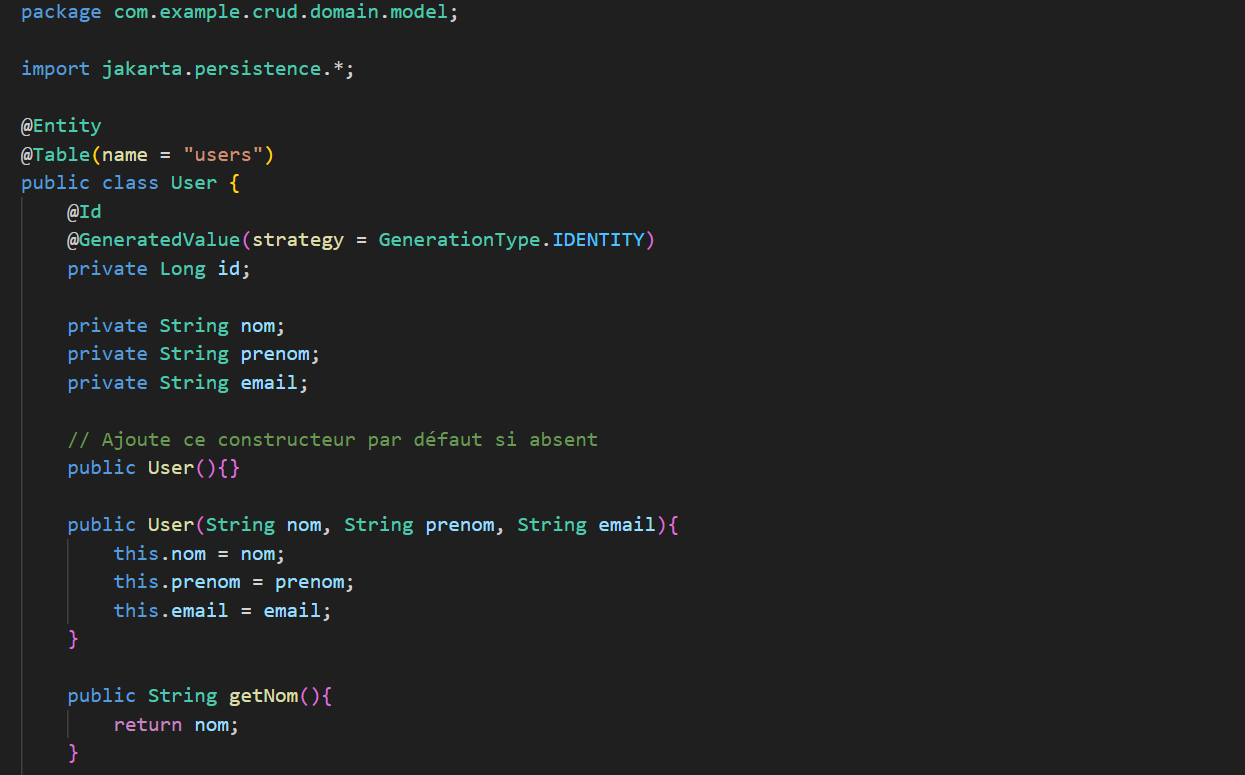
Cas MySQL :

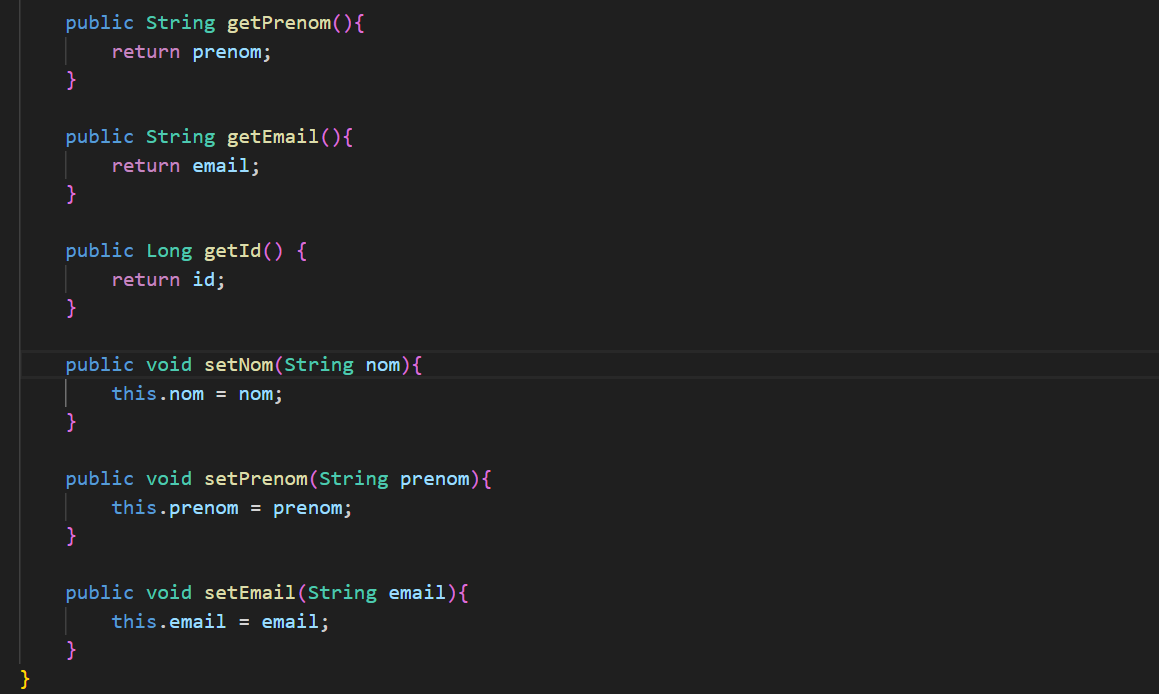


Cas MongoDB :

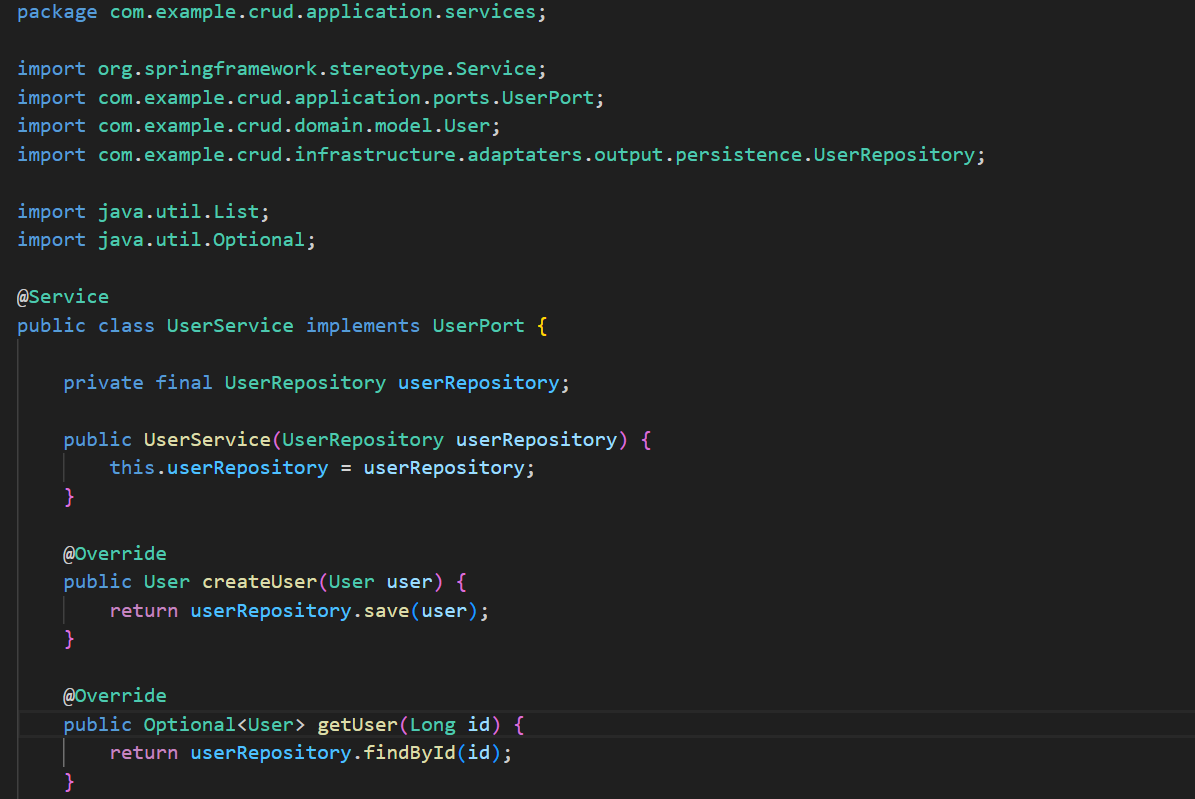


On crée notre model dans le repertoire com/example/crud/domain/model

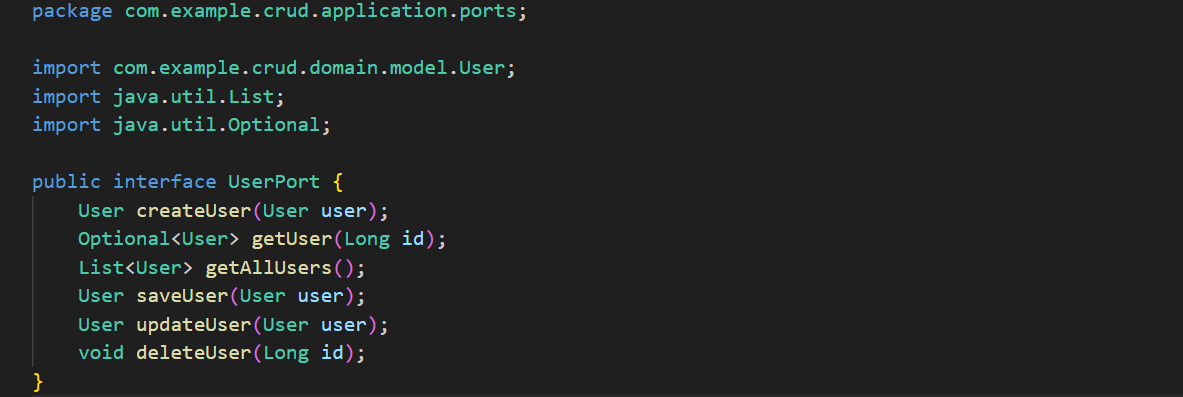




On crée notre service (UserService) com/example/crud/application/services

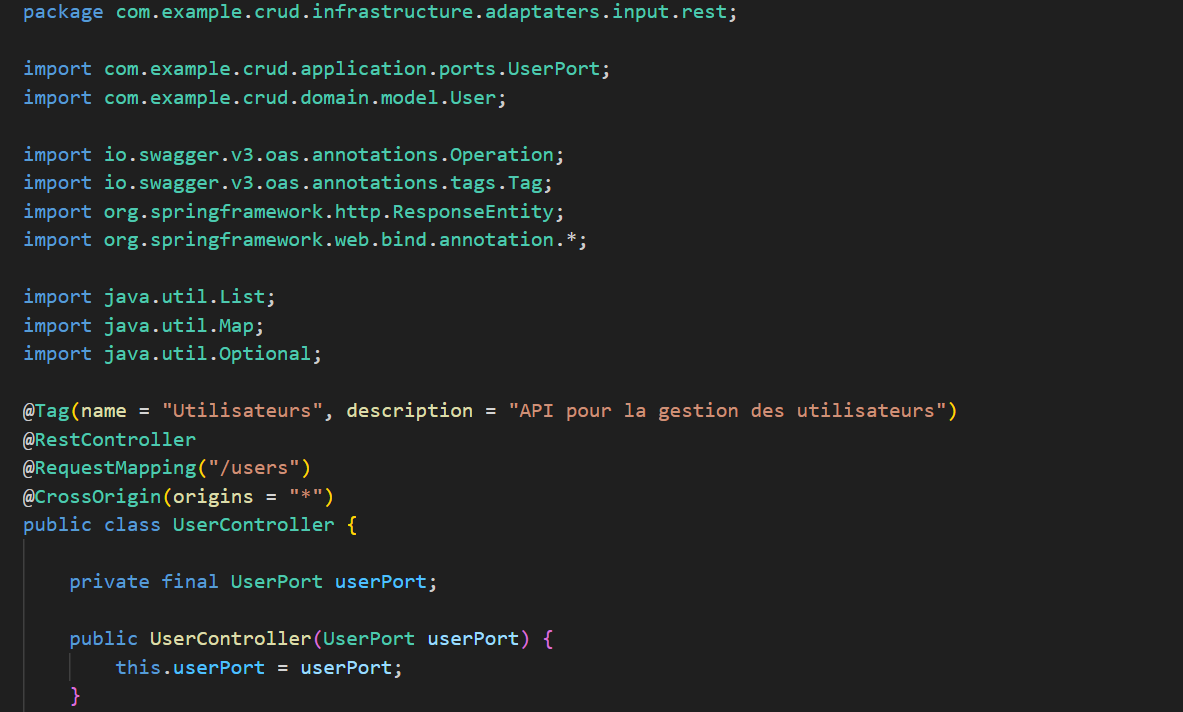


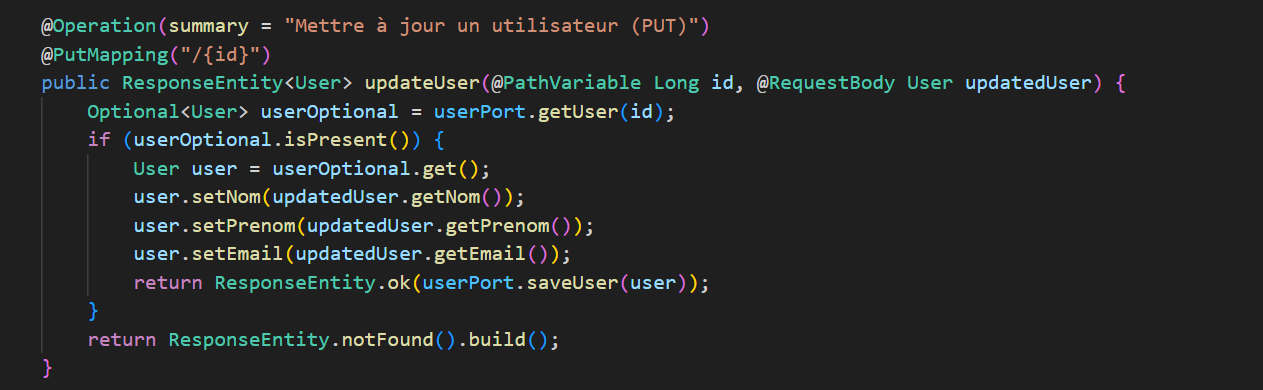
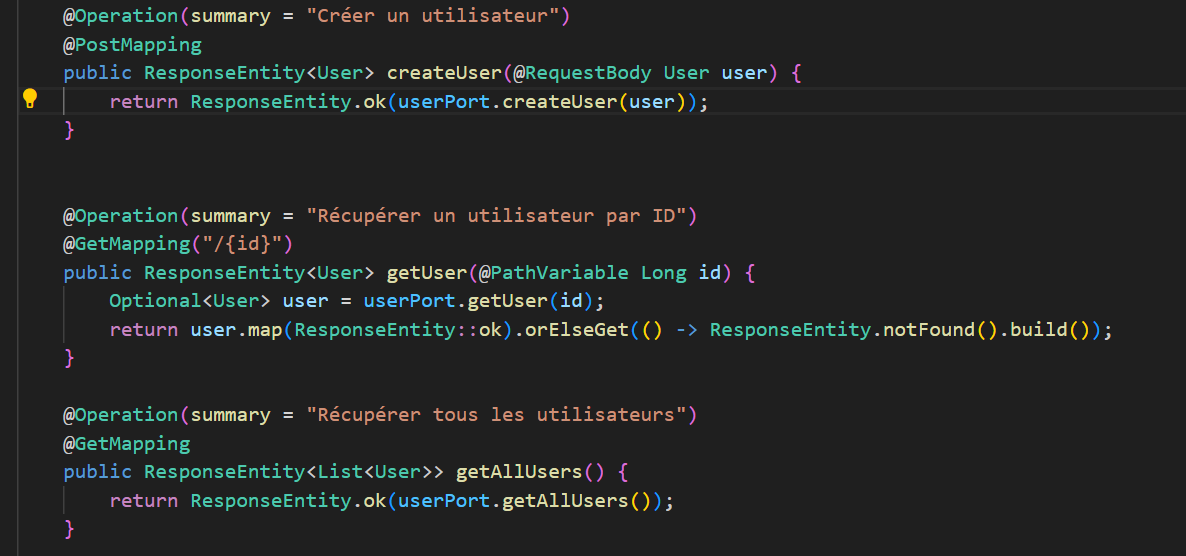
On créé notre port (UserPort) com/example/crud/application/ports

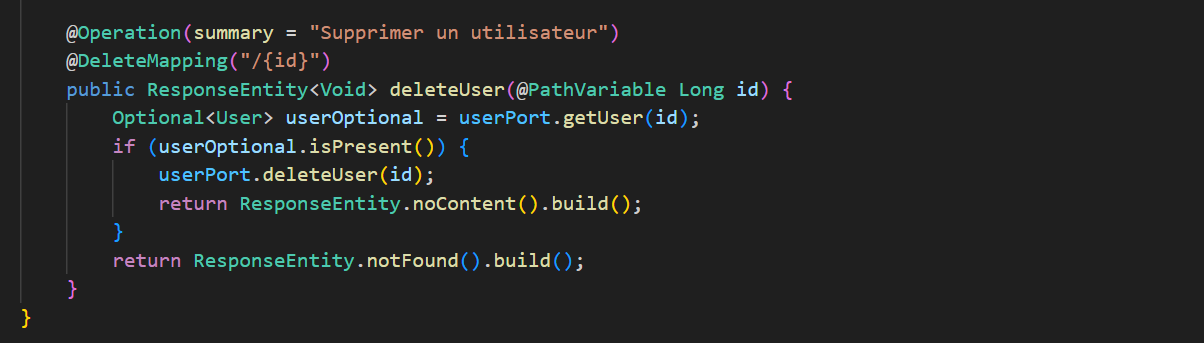


On crée notre com/example/crud/infrastructure/adaptaters

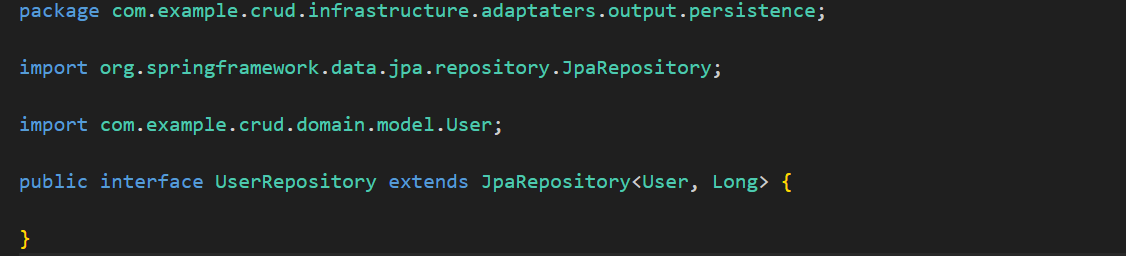
Dans infrastructure/adaptaters on crée le port input/rest dans lequel on va créer notre API REST



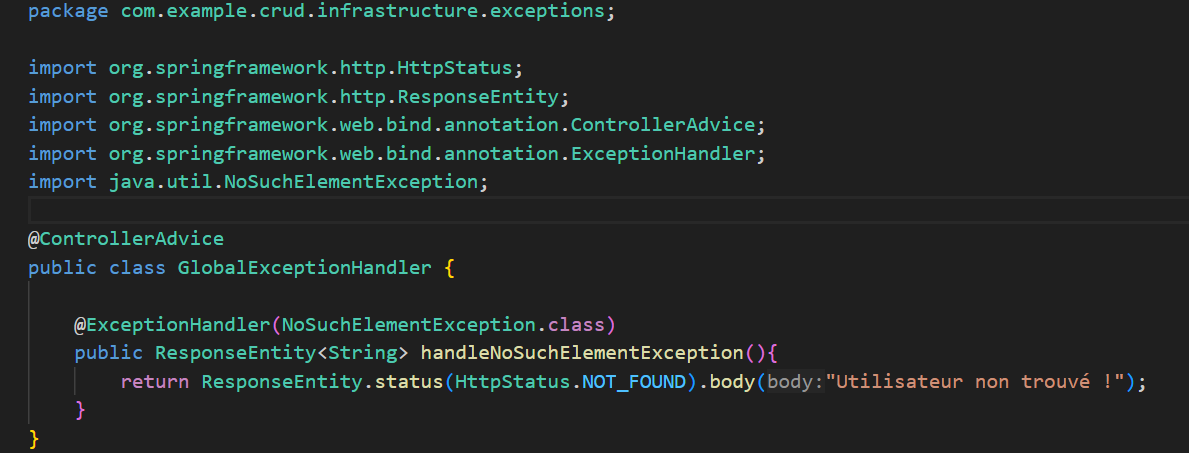




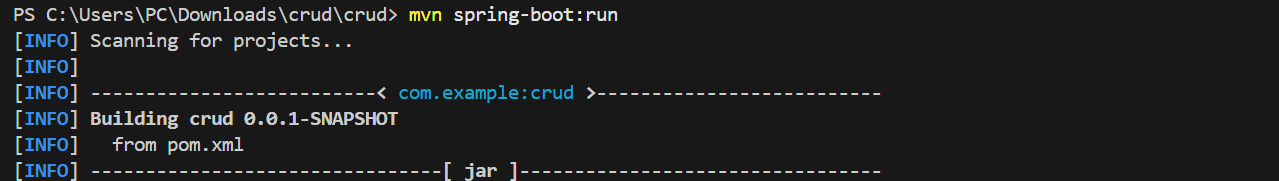
On crée notre port Output/persistence



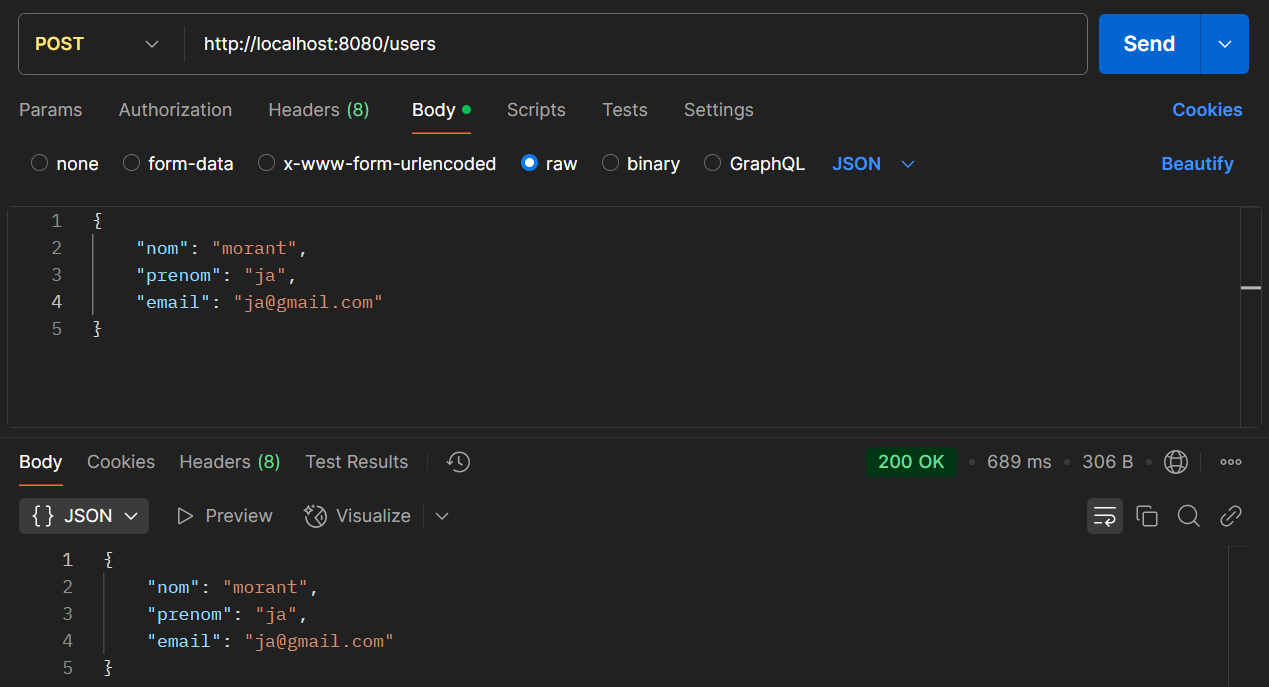
En même temps nous allons créer dans output les exceptions output/persistence pour gérer les messages d’erreur



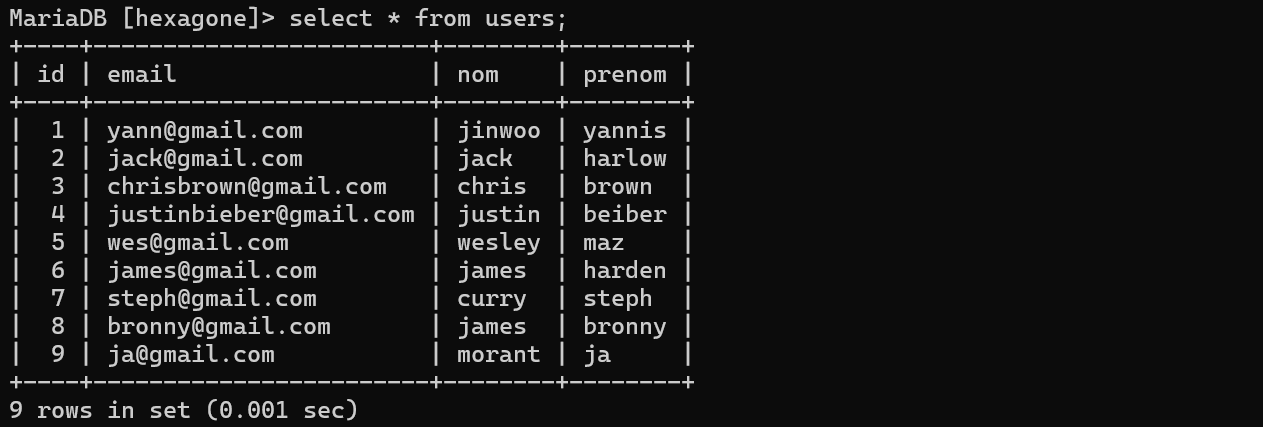
On démarre l’application mvn spring-boot:run



On va faire des enregistrements pour voir si notre api REST marche bien



On regarde dans la base de données si notre enregistrement marche



Regardons le rendu de notre api rest sur le sur web <http://localhost:8080/users>

