**Réalisé par : Amina Dahmouni**

**Classe : II-BDCC3**

**Encadré par : Pr Mohamed YOUSSFI**



|  |
| --- |
| **[Compte Rendu]** |
| Mise en œuvre d'une architecture micro-services **er, Création et Exécution d’une Image Docker, Plugin Eclipse)** |
| Systèmes Distribués et Big Data Processing **II BDCC – Ingénierie Informatique, Big Data et Cloud Computing** |
|  |

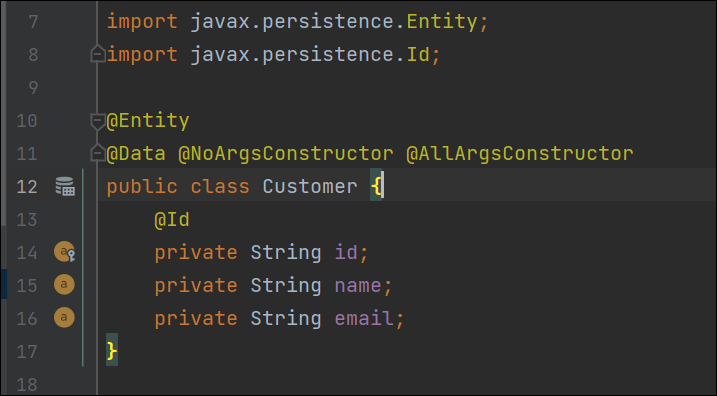




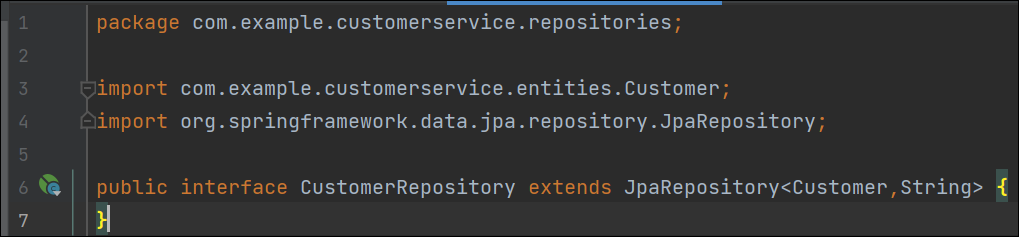
# Customer-Service

# Couche DAO (Entités JPA et Repositories)

## Entite Customer :

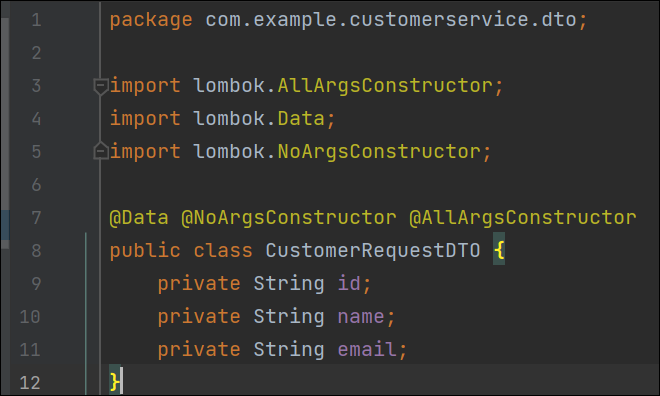


## CustomerRepository :

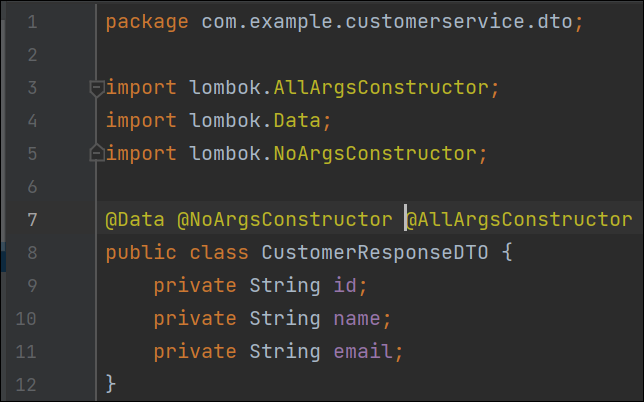


# La couche DTO

## CustomerRequestDTO :



## CustomerRespenseDTO :



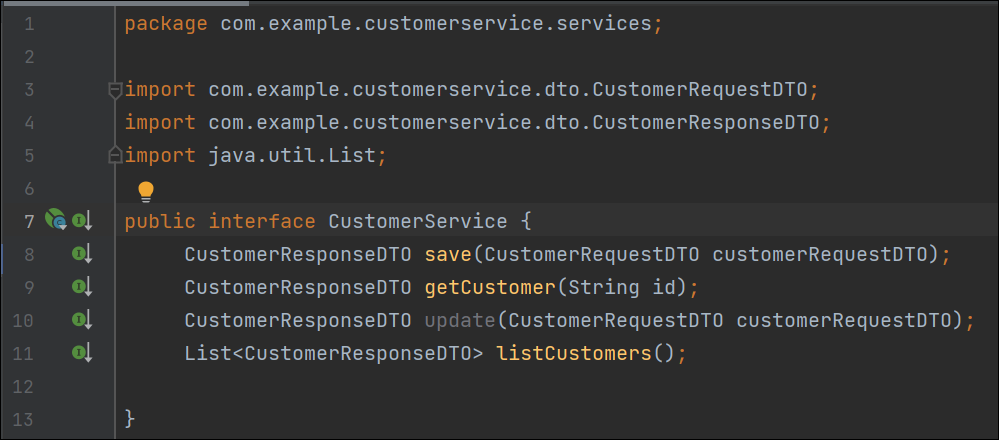
# Mappers (DTO <=>Entities)

## CustomerMapper :

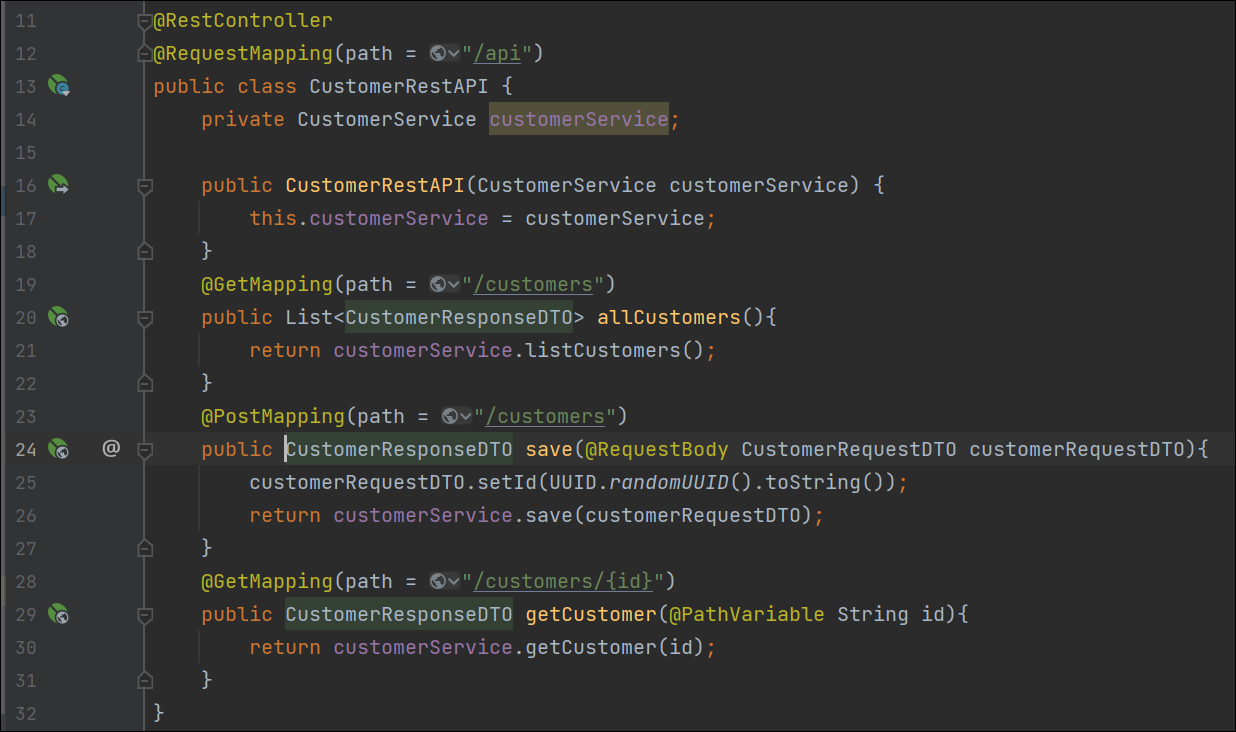


# Couche Service

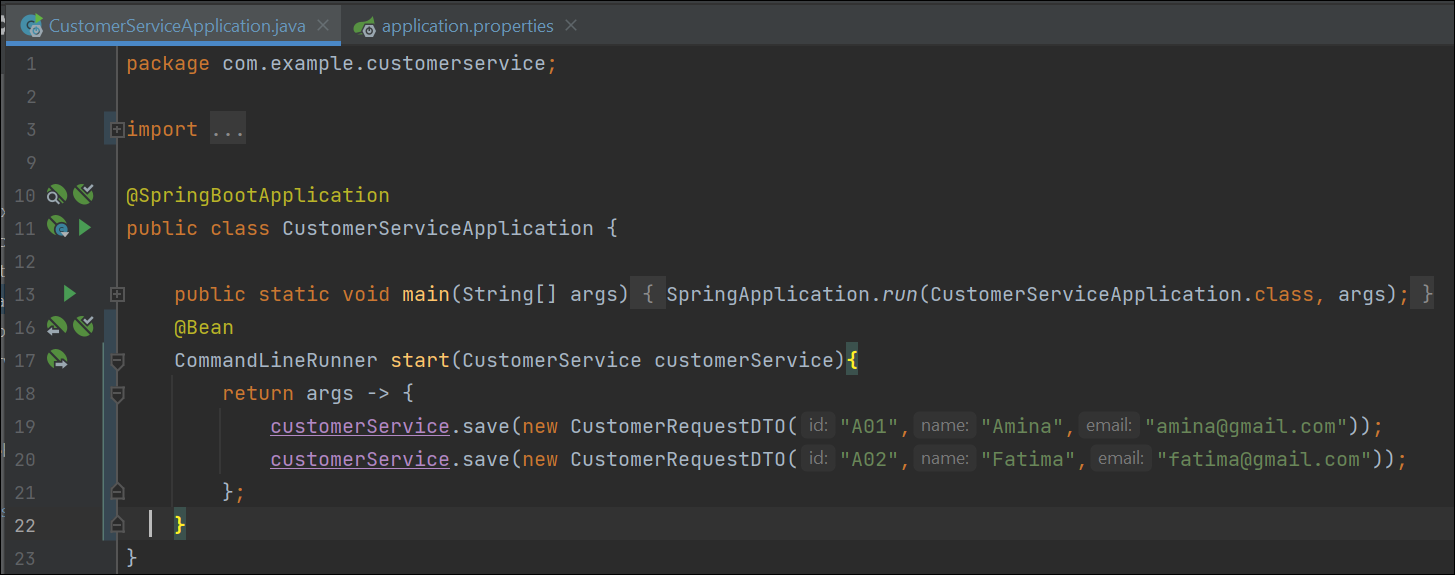
## CustomerService :



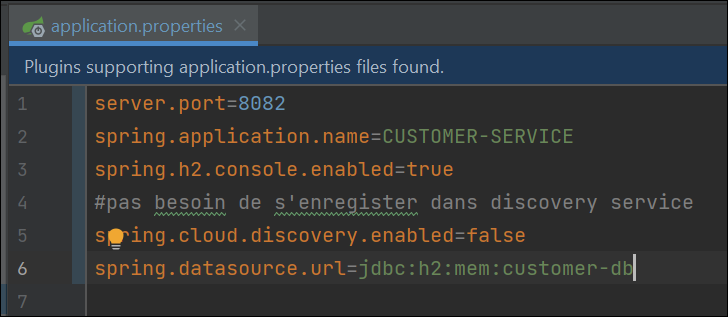
# La couche Web (Rest Controllers)



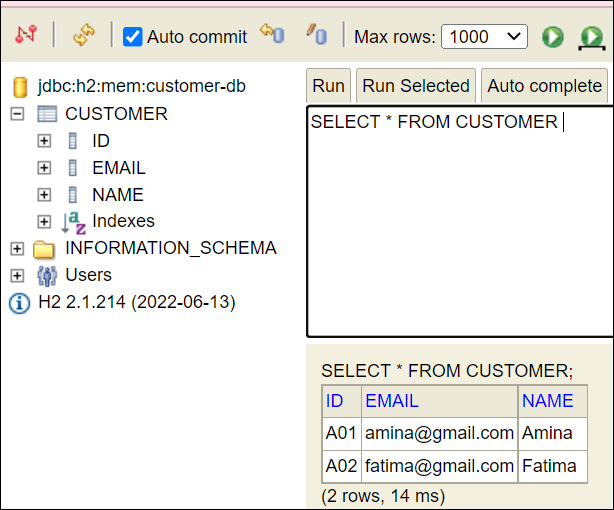
# Insertion des clients dans la base de données



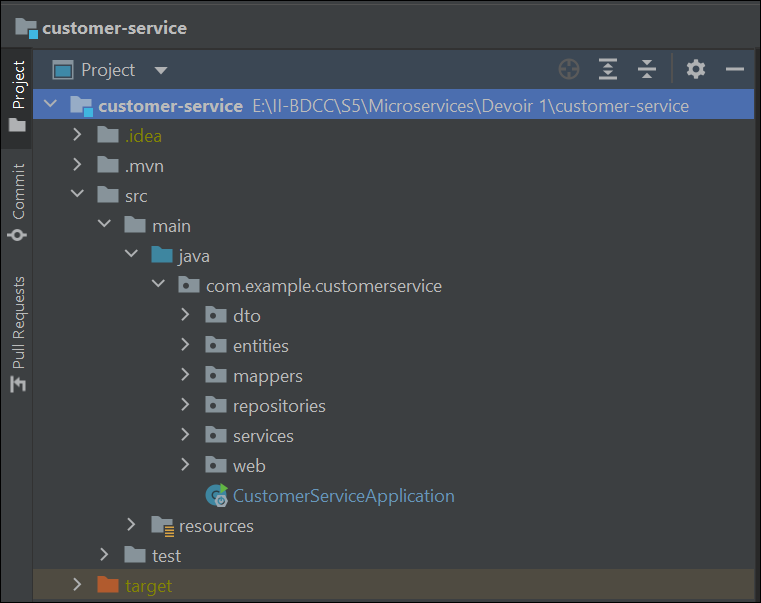
# Configuration de l’application



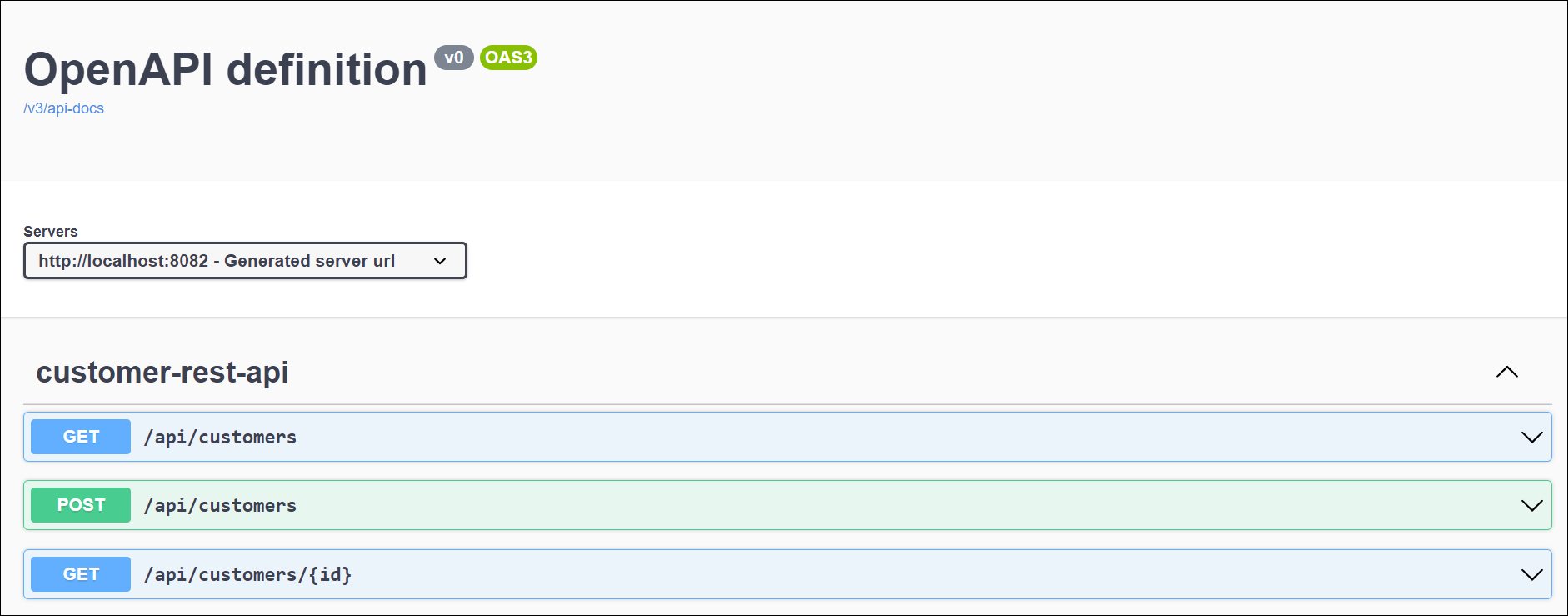
## Création de la base de données :

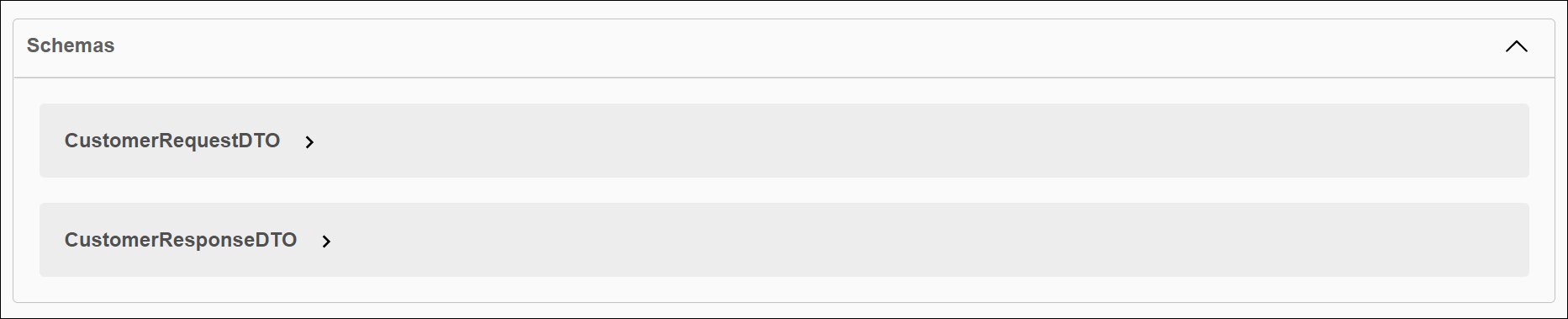


## La structure du projet :

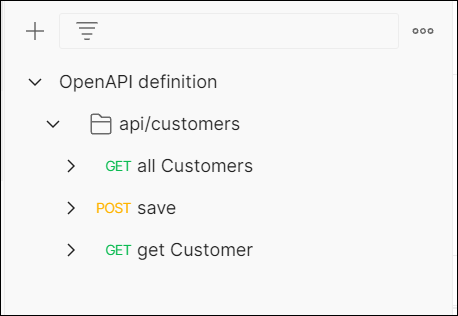


# La documentation du service avec Open API

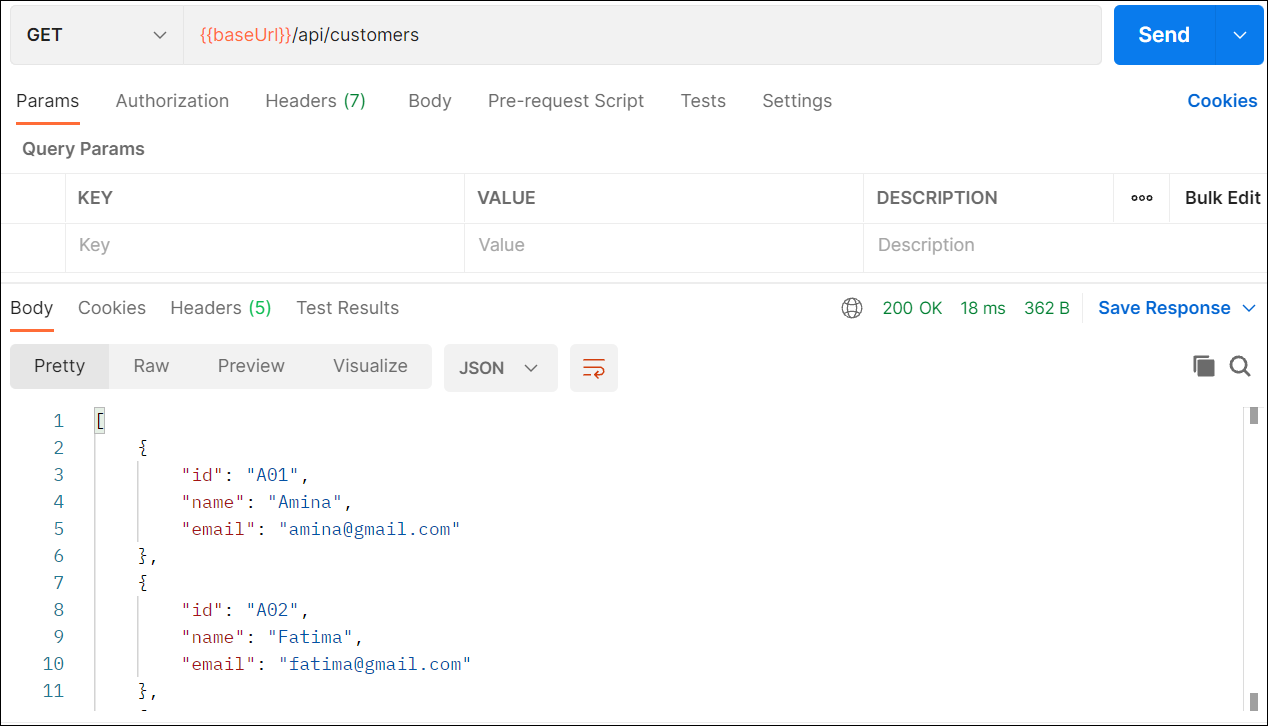




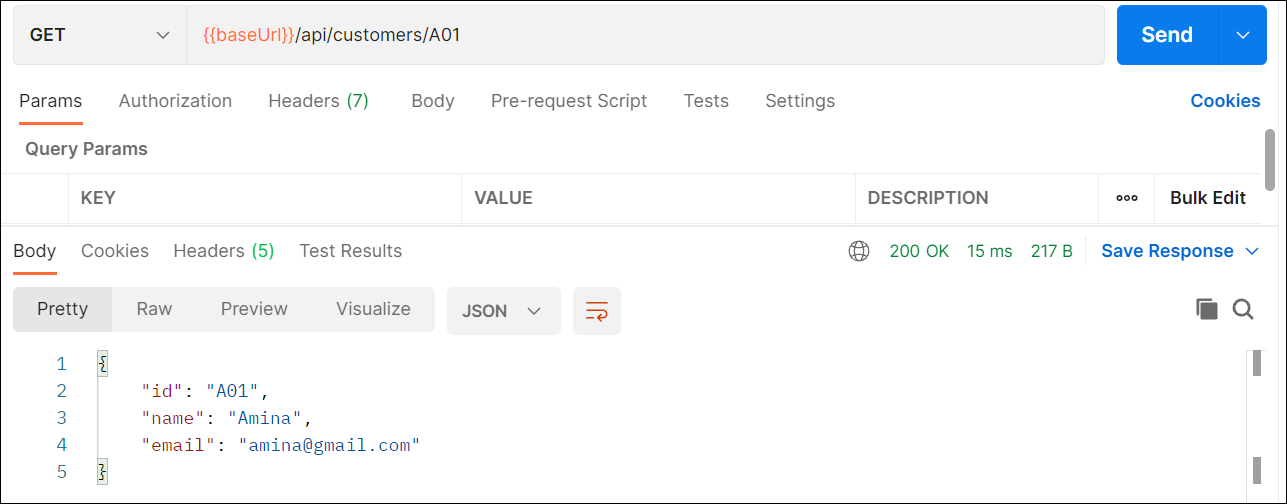
## Importer la documentation dans Postman :



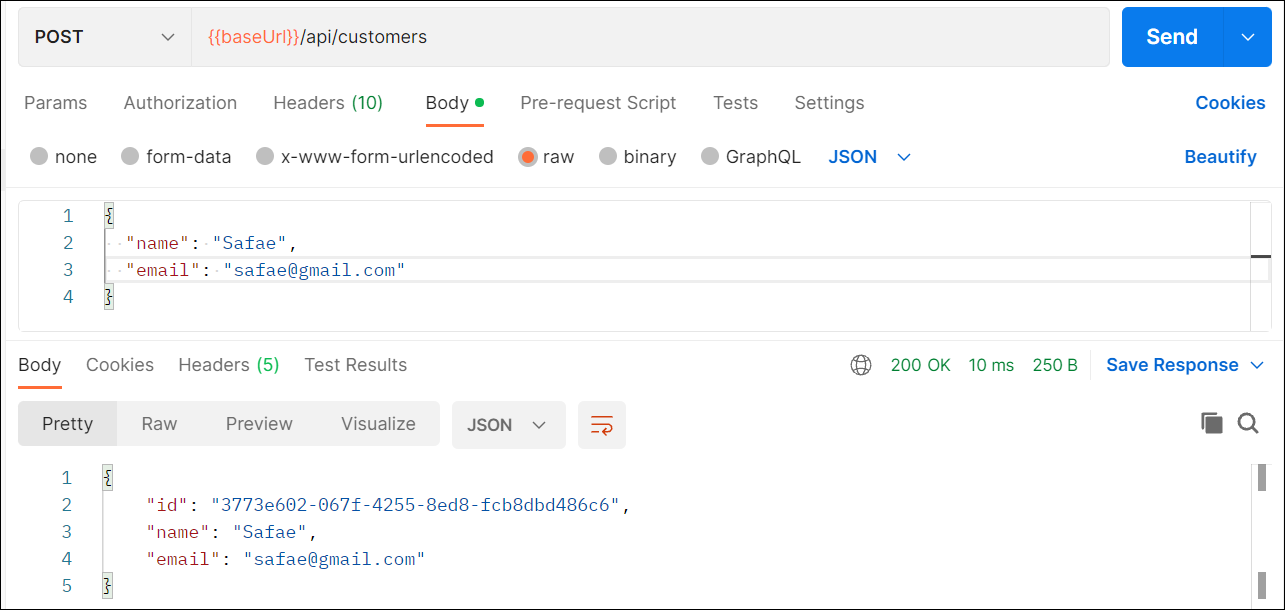
## Afficher la liste des clients :



## Chercher un client par ID :



## Ajouter un client :



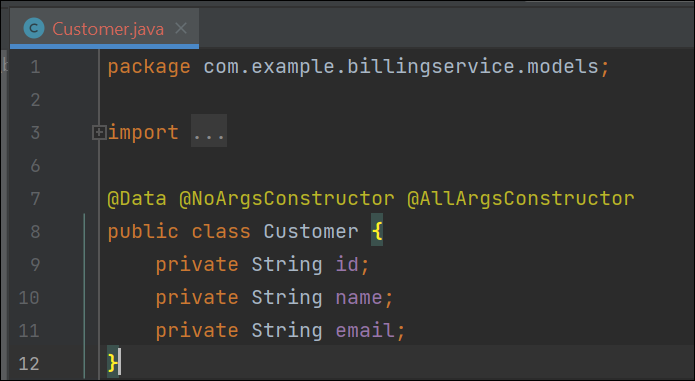


# Billing-Service

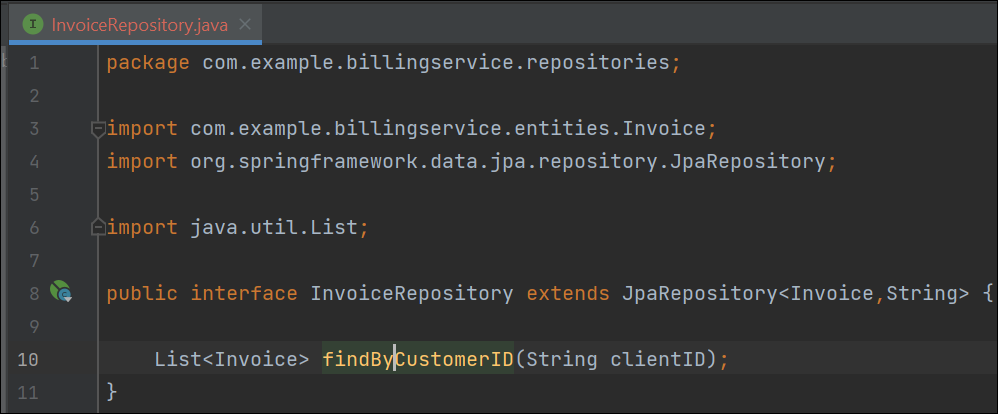
# Couche DAO (Entités JPA et Repositories)

## Entitie Invoice :



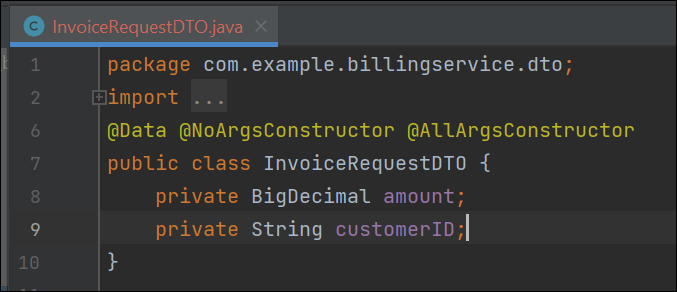


## InvoiceRepository :



# La couche DTO

## InvoiceRequest :

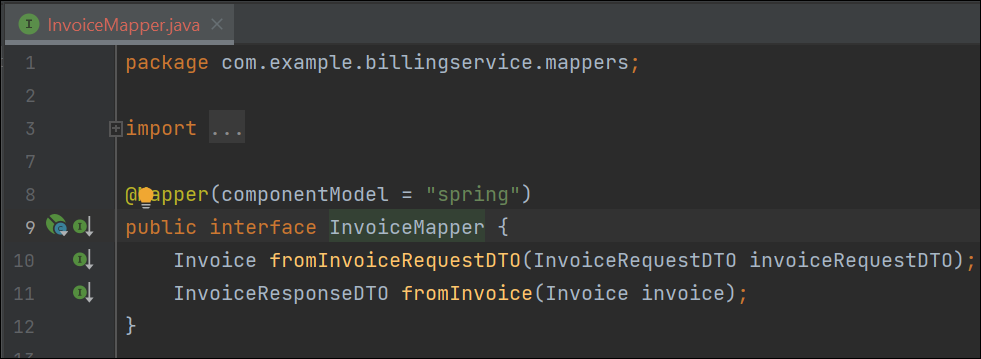


## InvoiceResponse :



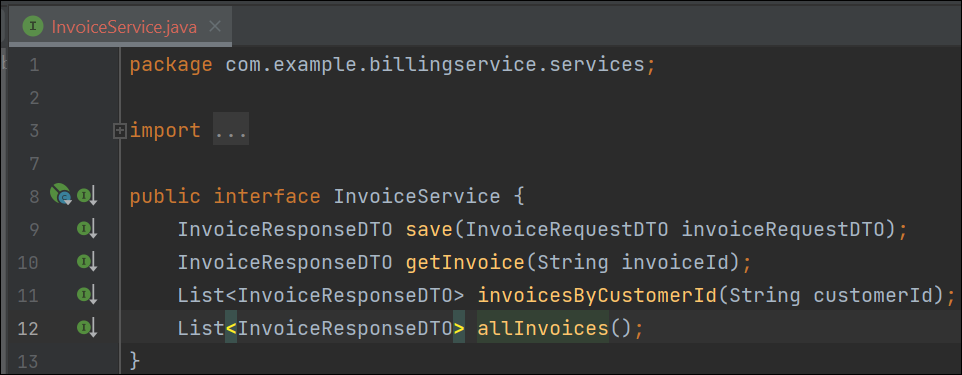
# Mappers (DTO <=>Entities)

## InvoiceMapper :



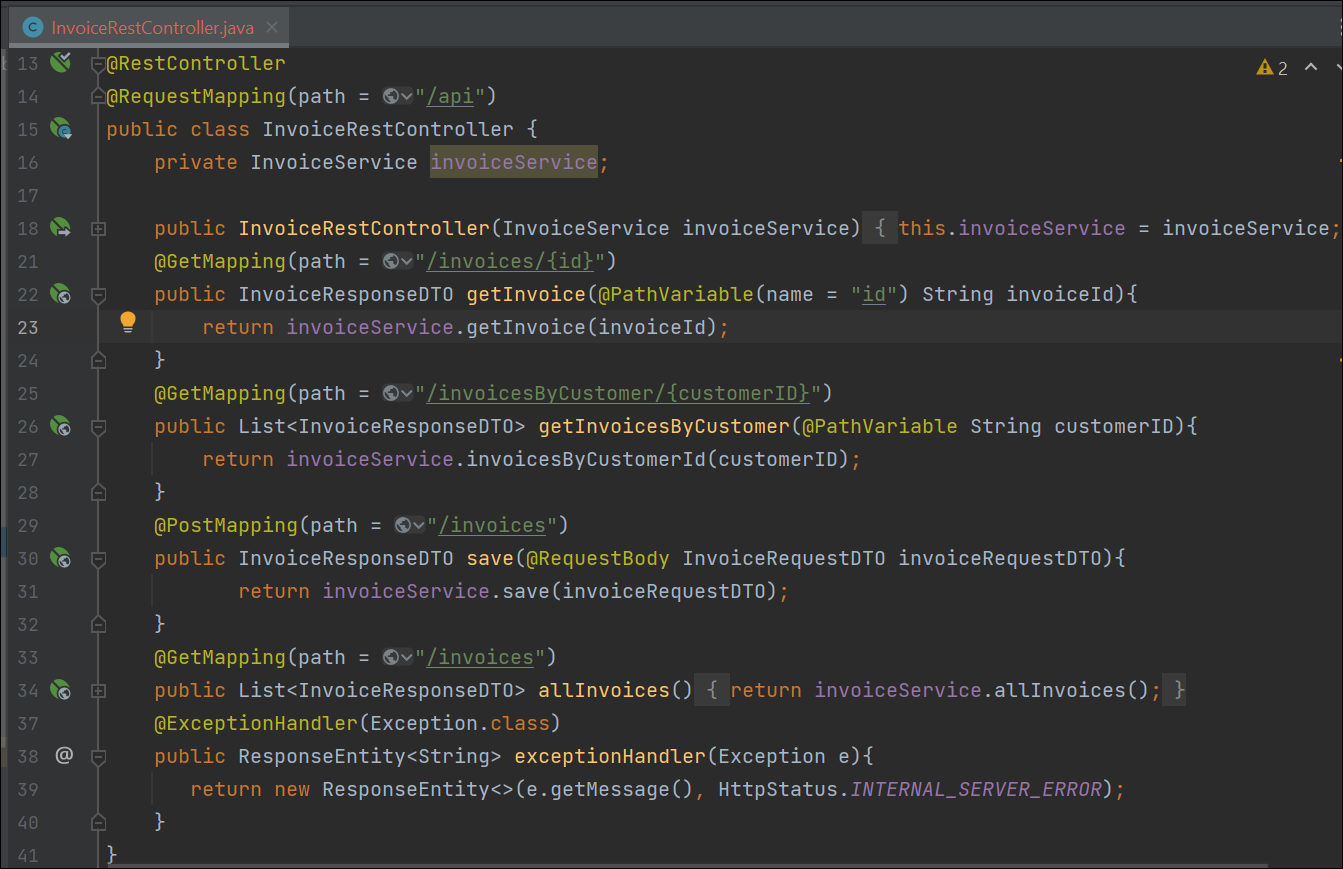
# Couche Service

## InvoiceService :



# La couche Web (Rest Controllers)

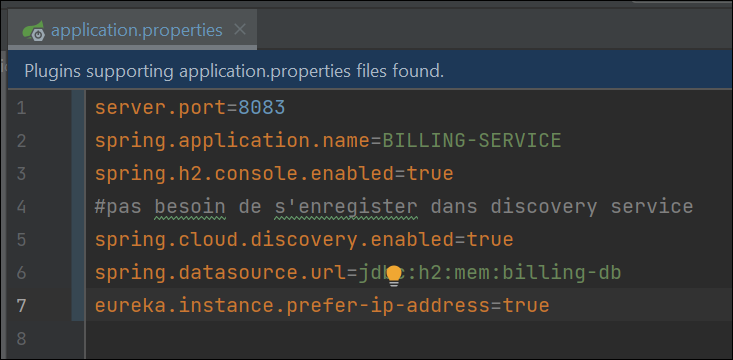
## InvoiceRestController :



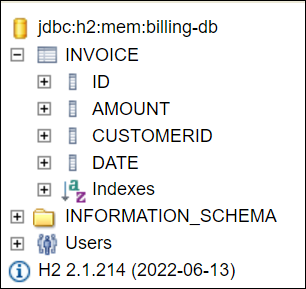
# Insertion des clients dans la base de données



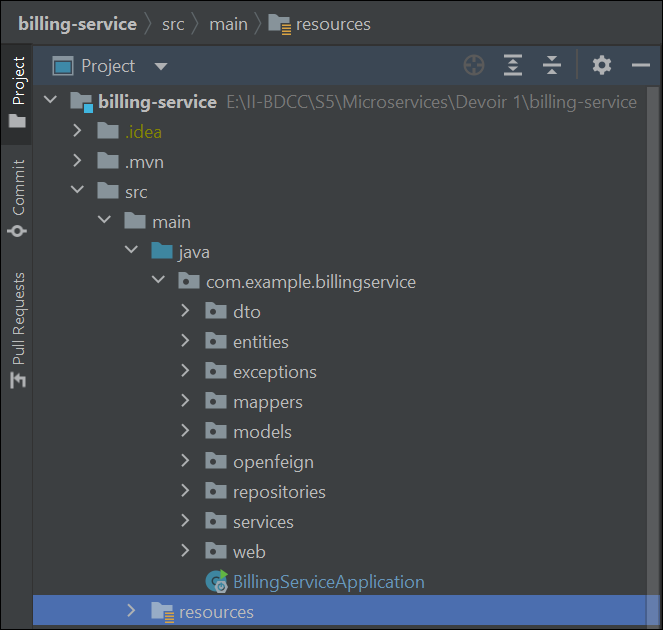
# Configuration de l’application



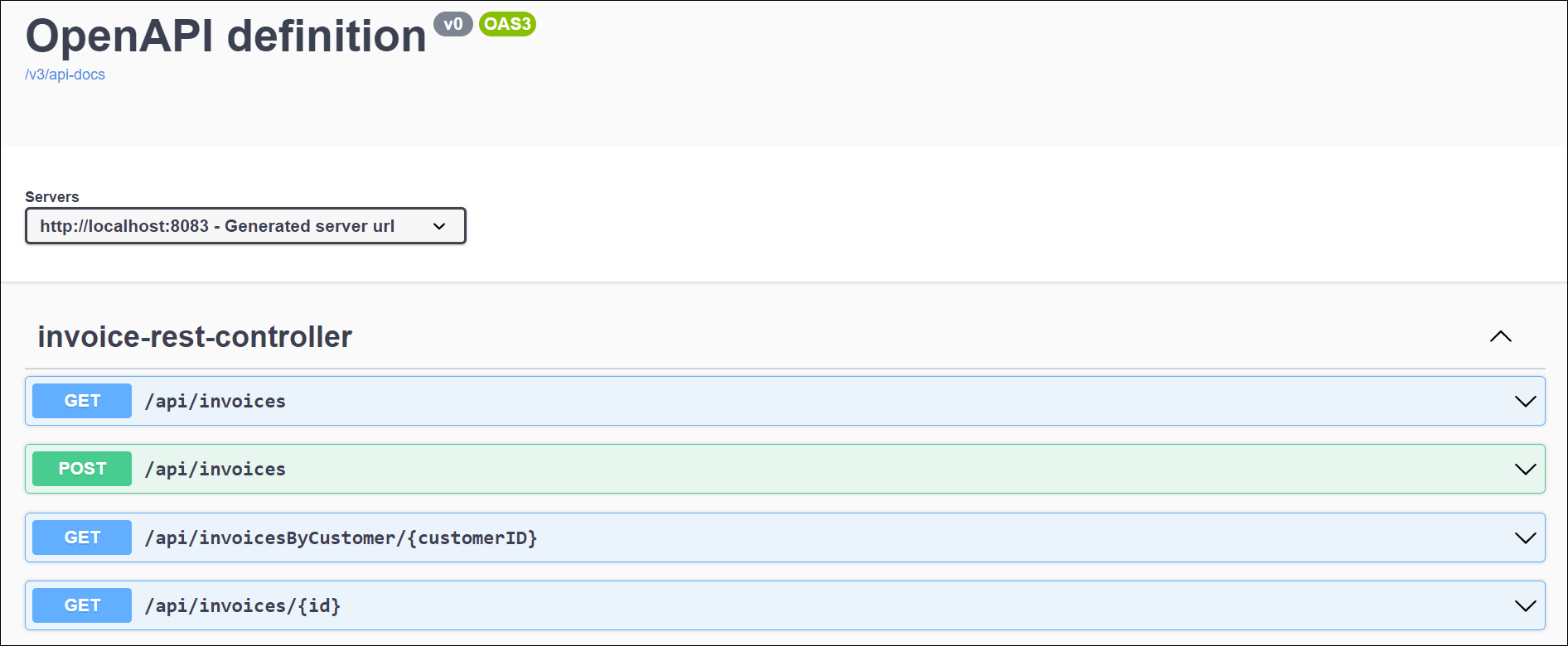
## Création de la base de données :

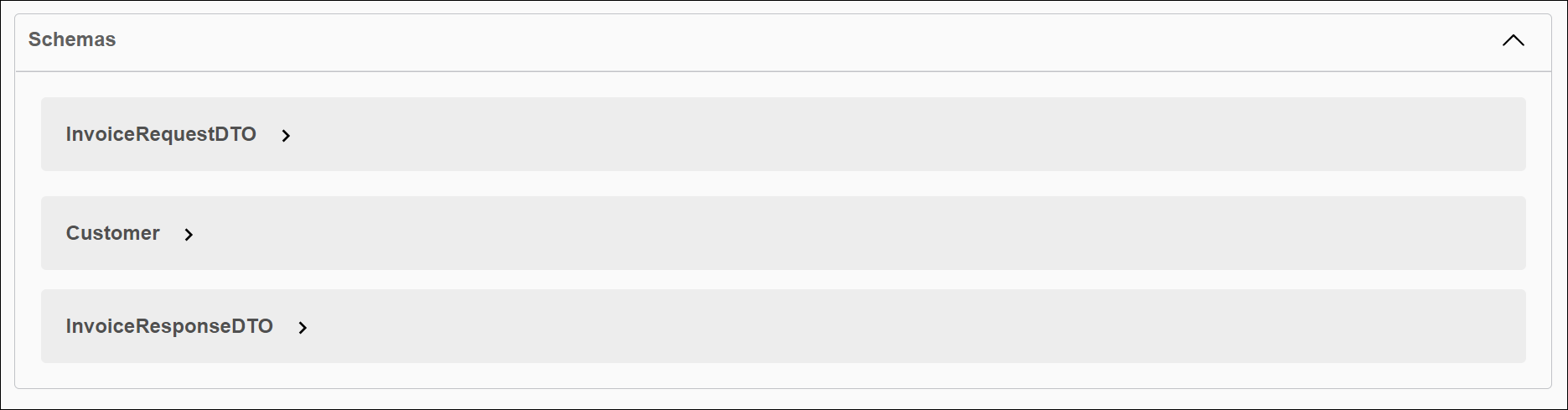


## La structure du projet :

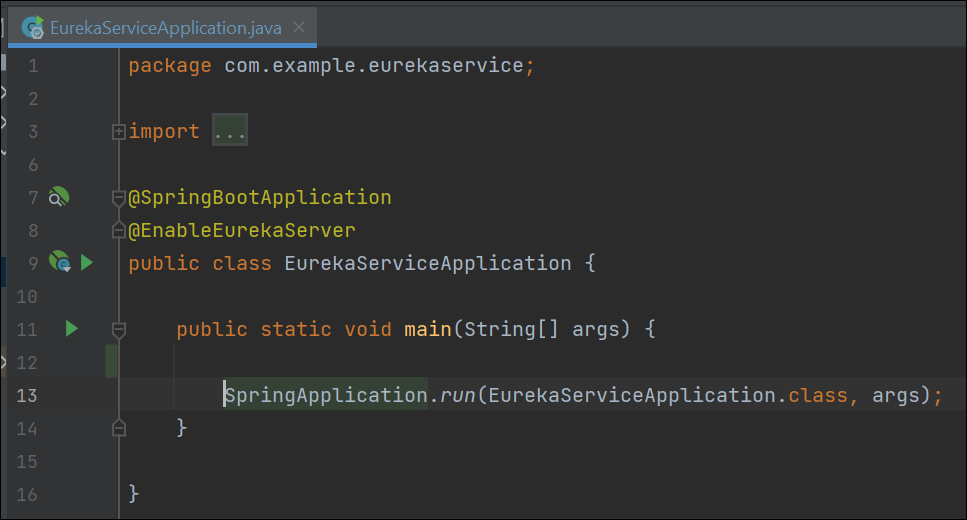


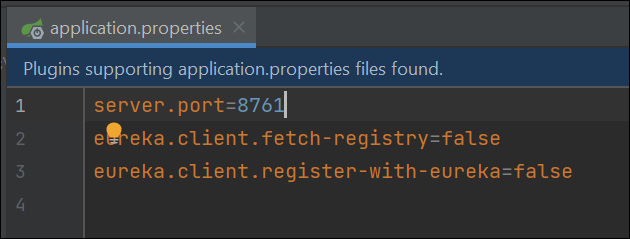
# La documentation du service avec Open API

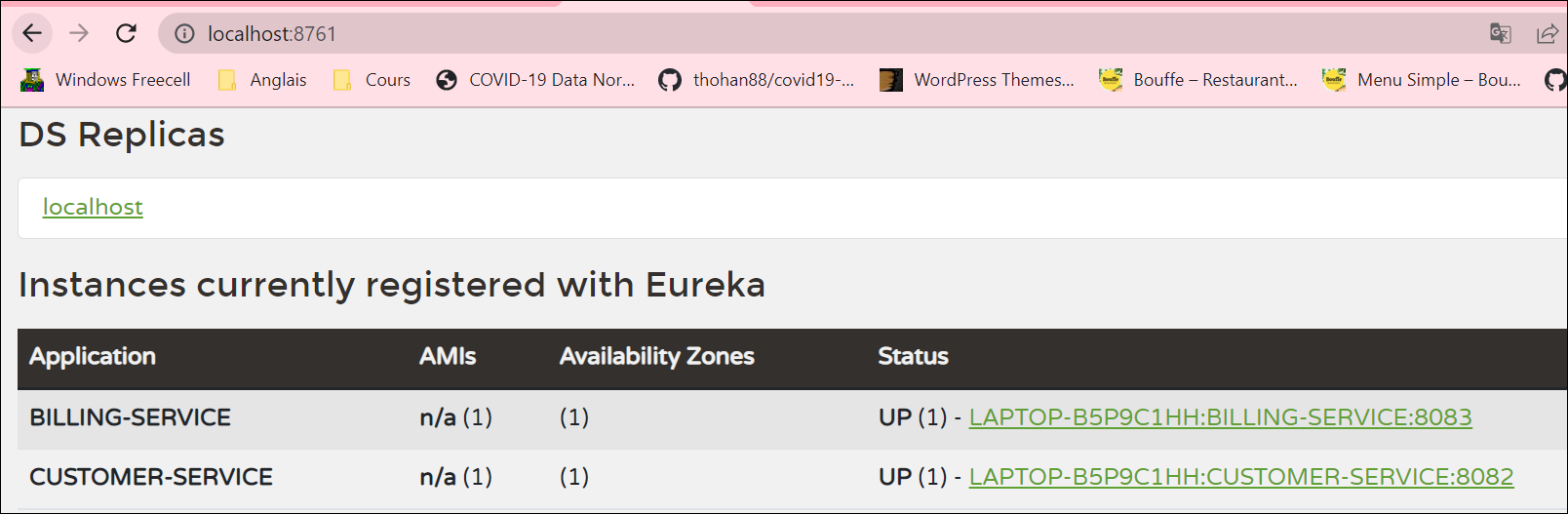




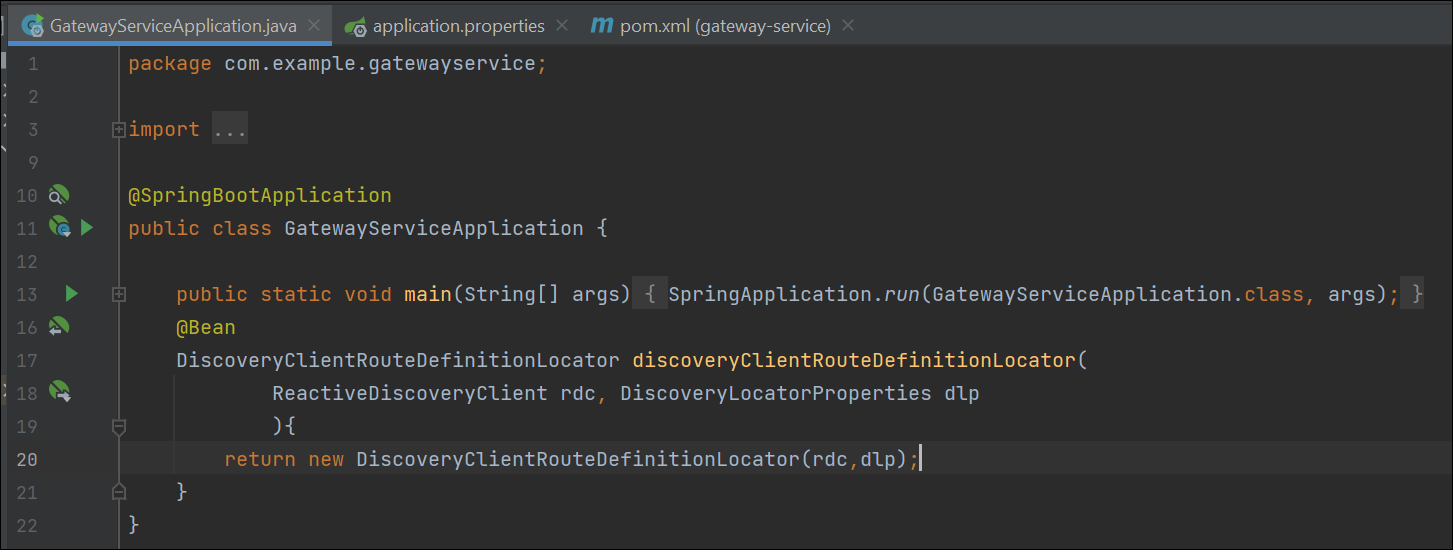
# Eureka-Service

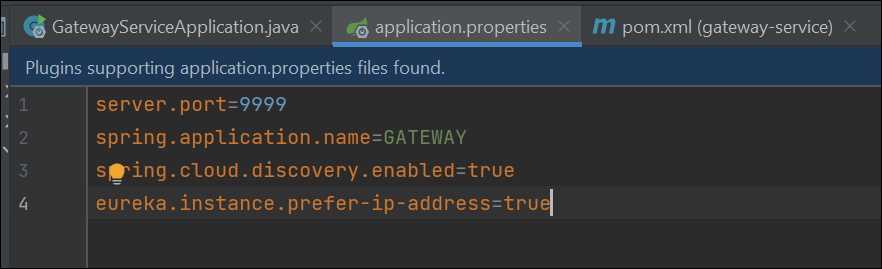


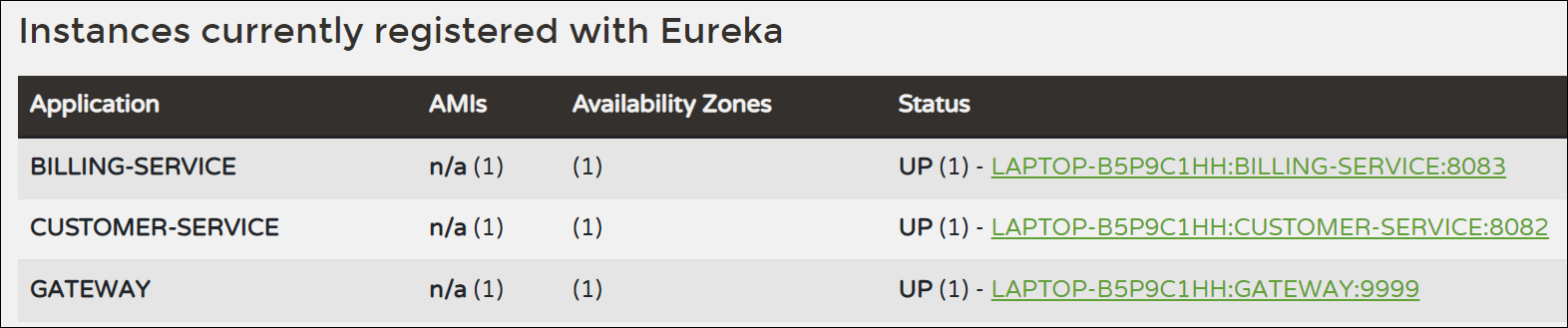




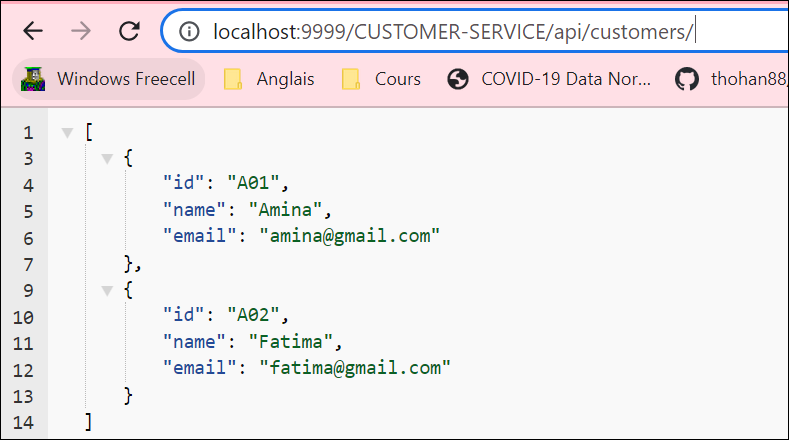
# Gateway-Service



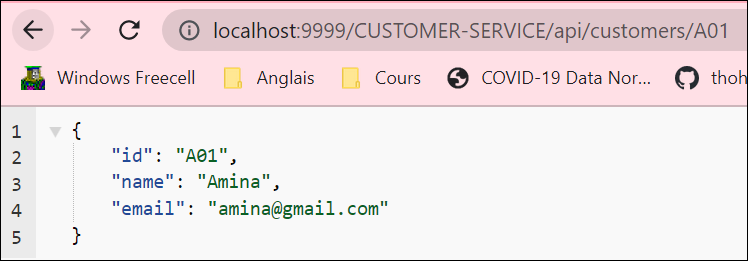




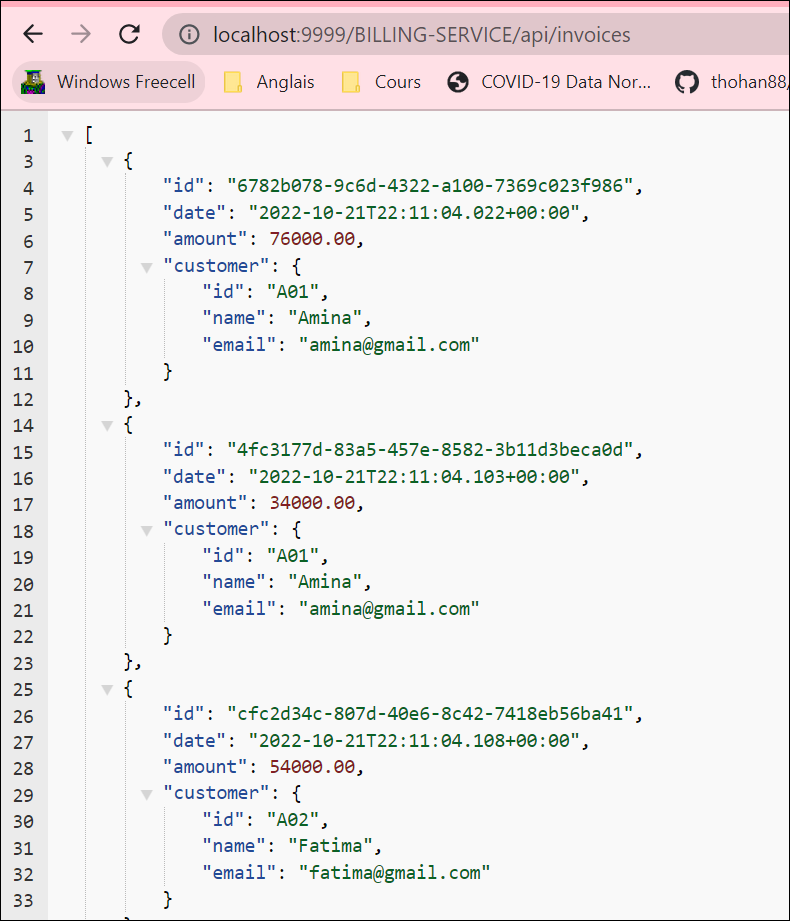
## Afficher la liste des Clients :

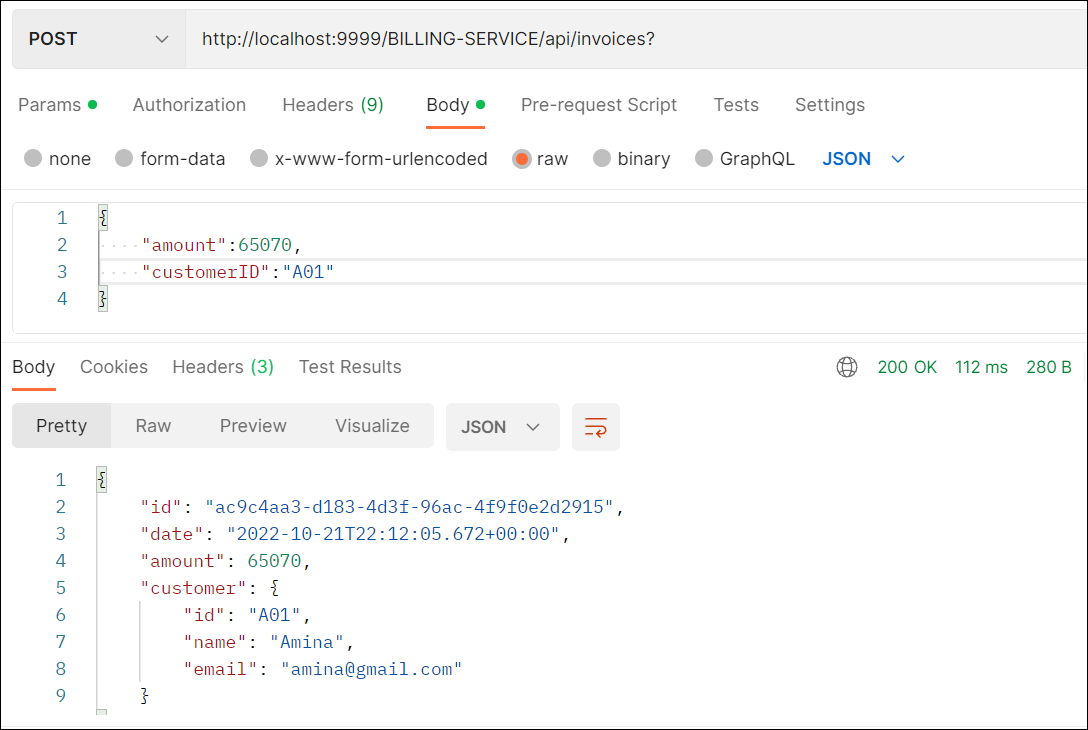


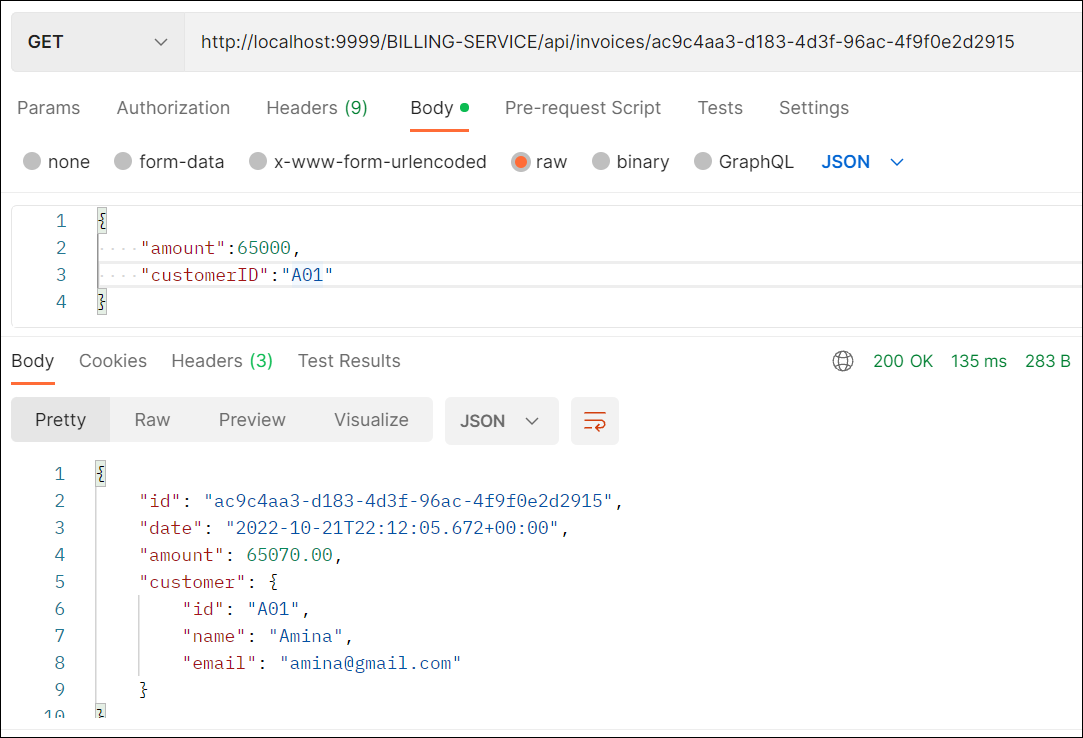
## Chercher un client par ID :



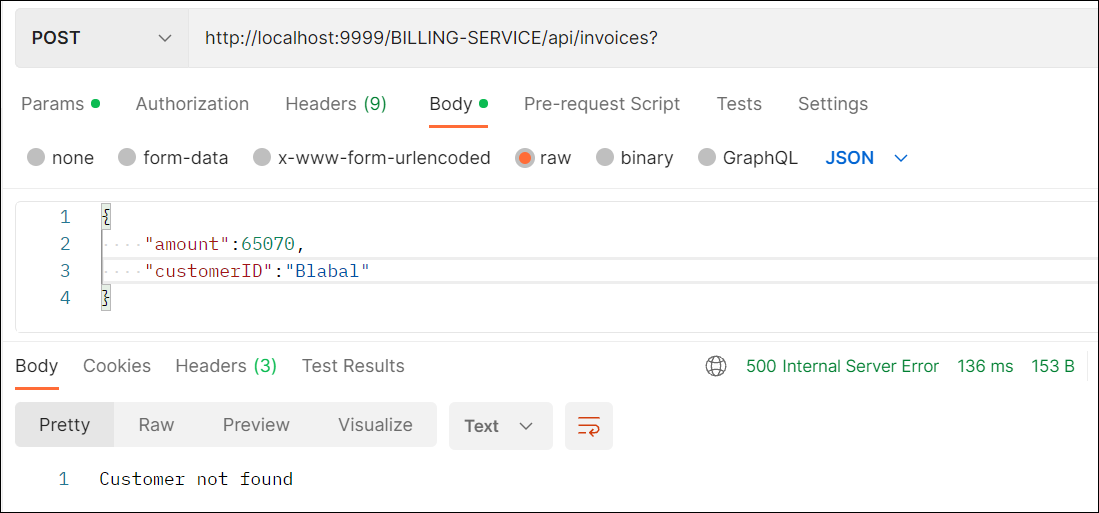
## Afficher la liste des factures :







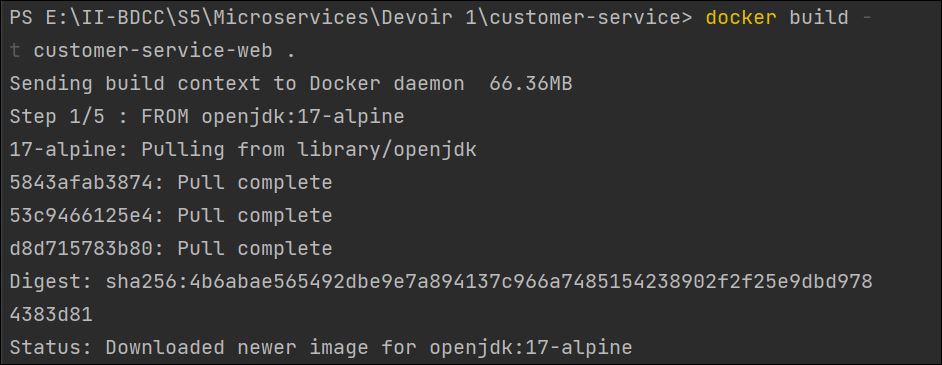
## Ajouter une facture à un client qui n‘existe pas :



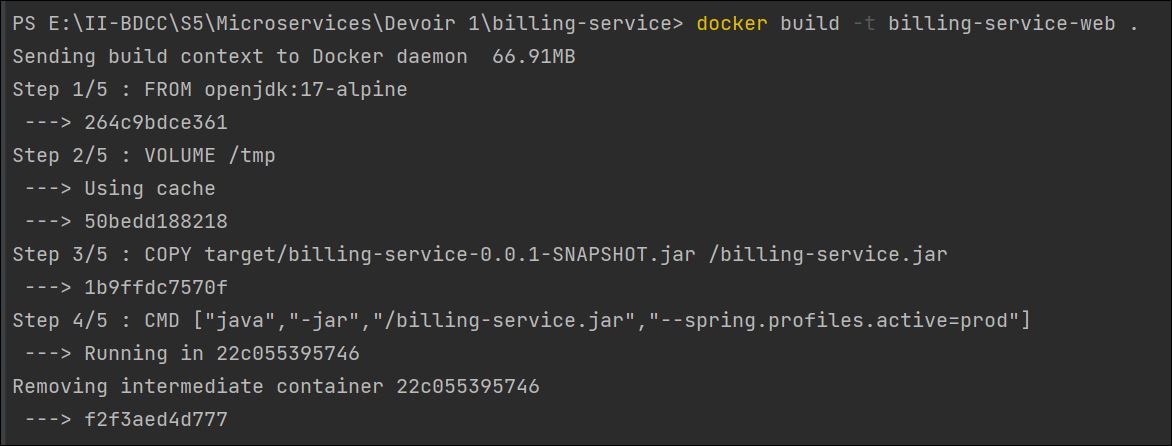
Il génère l’exception CustomerNotFound.

**Déployer les micro services de démarrage à ressort dans un conteneur docker**

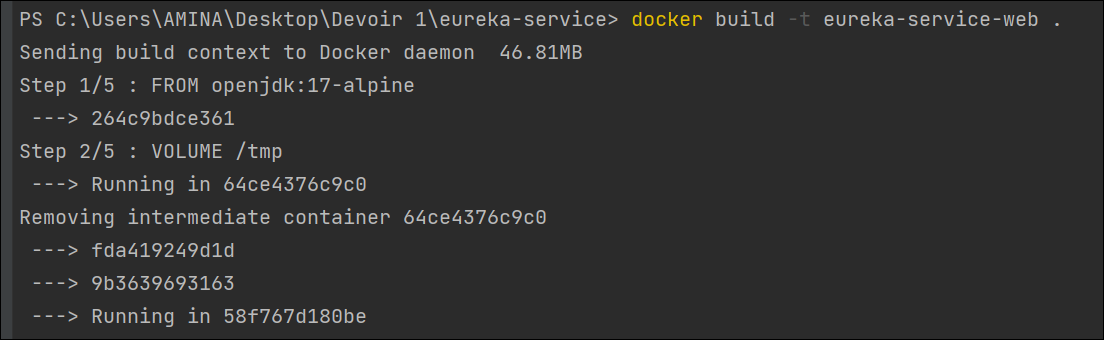
Déployer customer-service :



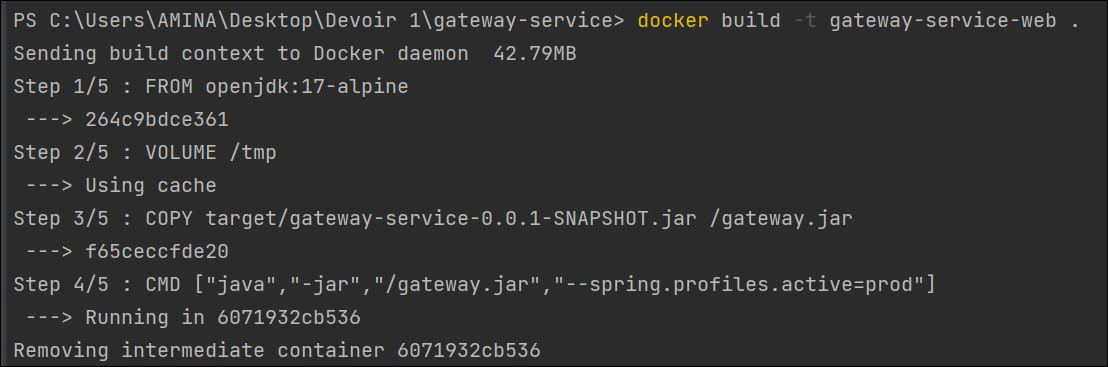
Déployer billing-service :



Déployer eureka-service :



Déployer gateway-service :



Créer un nouveau fichier docker-compose.yml pour exécuter, à la fois, les image des micro services :

