ECOLE NORMALE SUPÉRIEURE DE
SEIGNEMENT TECHNIQUE DE MOHAMMEDIA

UNIVERSITÉ HASSAN II DE CASABLANCA

ENSET

المدرسة العليا لأساتذة التعليم التقني
المحمدية
جامعة الحسن الثاني بالدار البيضاء

DÉPARTEMENT MATHÉMATIQUES ET INFORMATIQUE

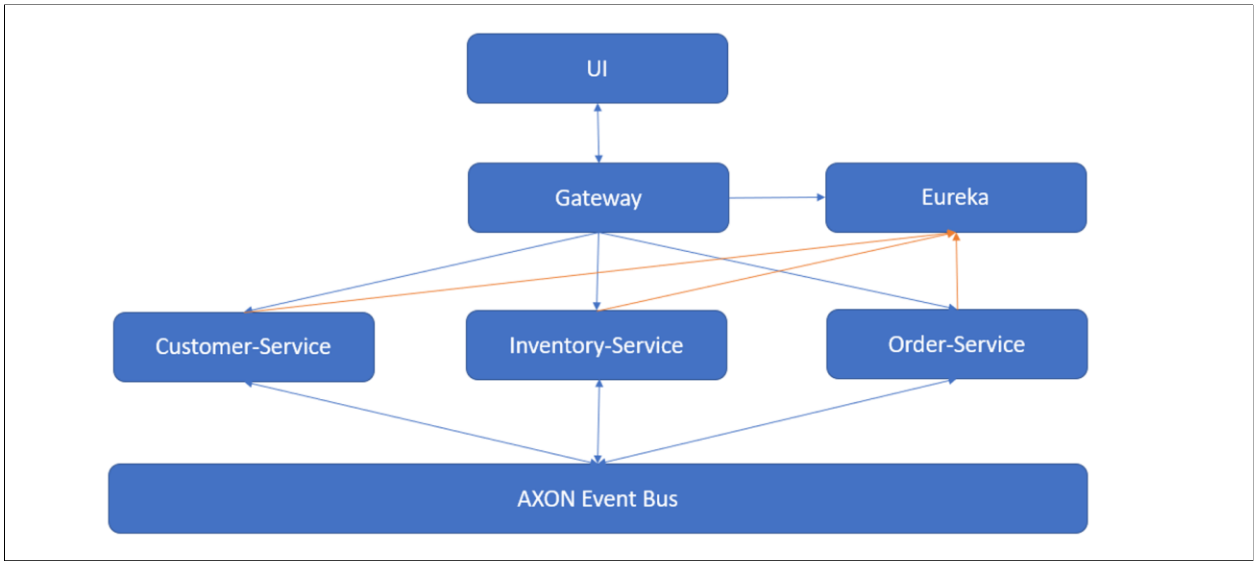
RAPPORT EXAM MICROSERVICES :

Réaliser Par :
Lahcen LAMKIRICH

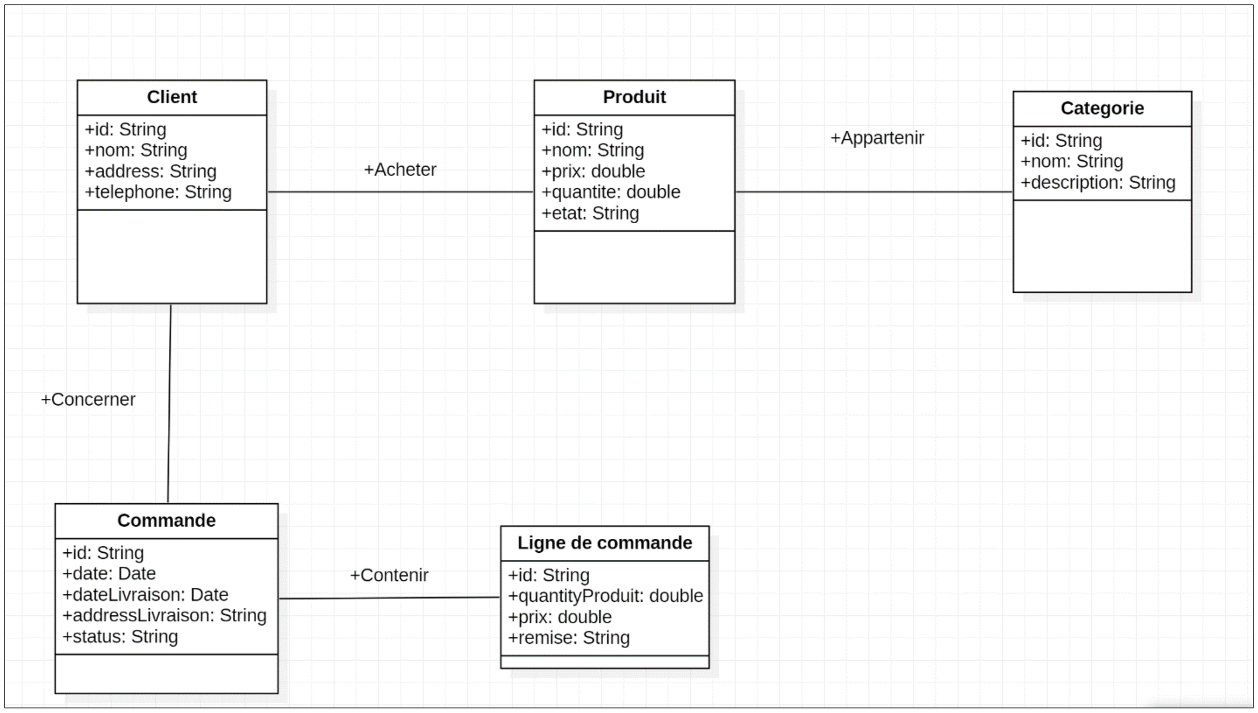
Encadré Par :
M. Mohamed YOUSSFI

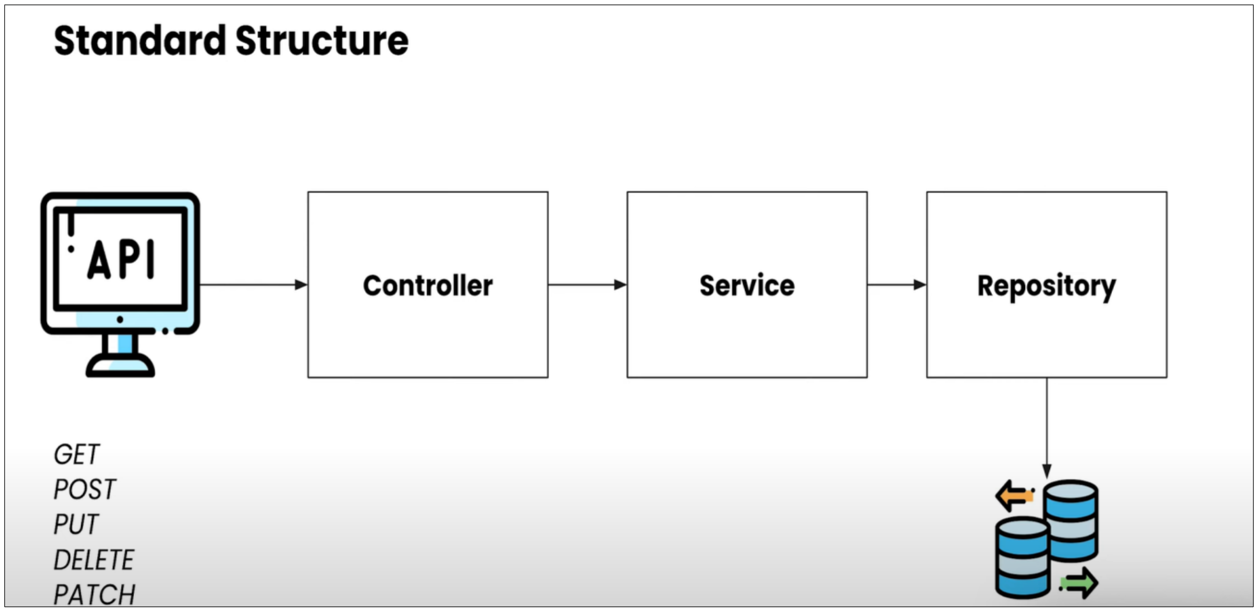
ANNEE SCOLAIRE :
2022/2023

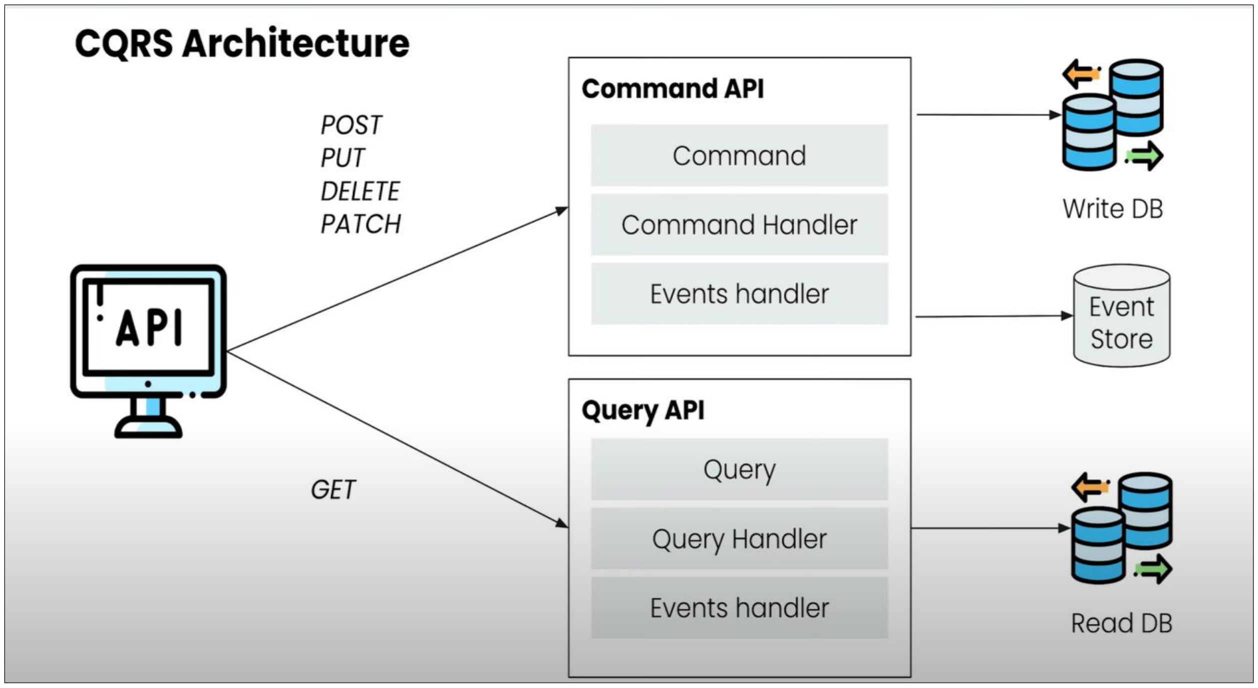
1. Établir une architecture technique du projet

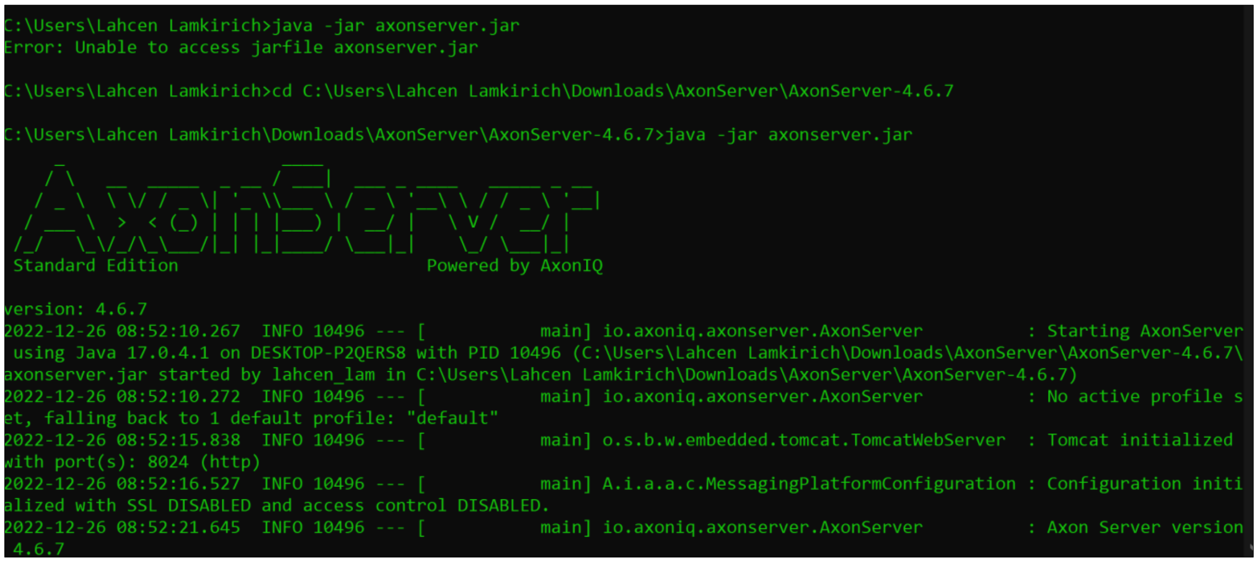


2- Établir un diagramme de classe global du projet

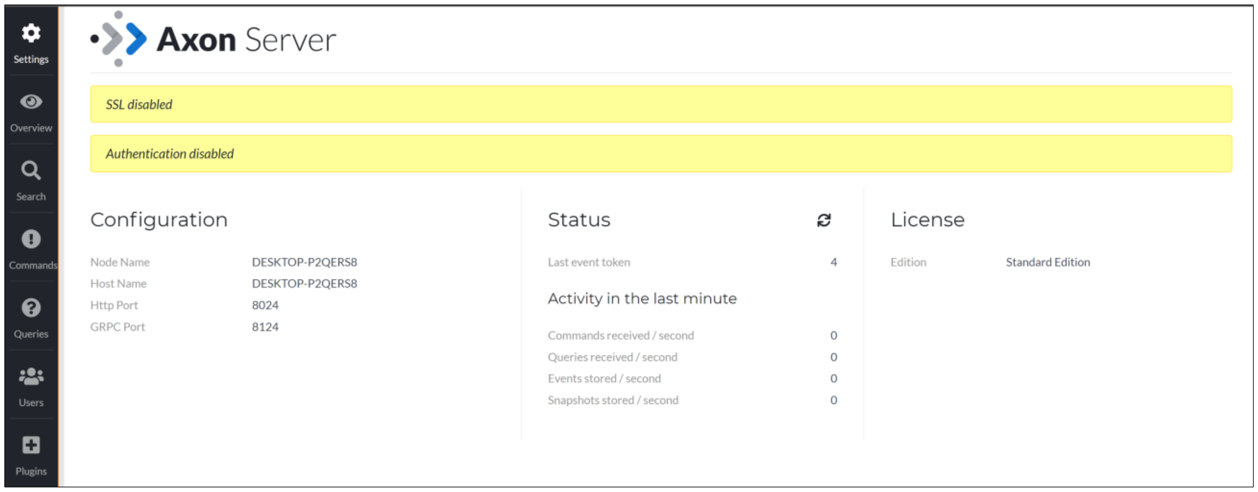



Standard Structure
API
Controller
Service
Repository
GET
POST
PUT
DELETE
PATCH


CQRS Architecture
POST
PUT
DELETE
PATCH
API
GET
Command API
Command
Command Handler
Events handler
Query API
Query
Query Handler
Events handler
Write DB
Event Store
Read DB



```
C:\Users\Lahcen Lamkirich>java -jar axonserver.jar
Error: Unable to access jarfile axonserver.jar

C:\Users\Lahcen Lamkirich>cd C:\Users\Lahcen Lamkirich\Downloads\AxonServer\AxonServer-4.6.7

C:\Users\Lahcen Lamkirich\Downloads\AxonServer\AxonServer-4.6.7>java -jar axonserver.jar
     _                     ____
    / \   __  _____  _ __ / ___|  ___ _ ____   _____ _ __
   / _ \  \ \/ / _ \| '_ \\___ \ / _ \ '__\ \ / / _ \ '__|
  / ___ \  >  < (_) | | | |___) |  __/ |   \ V /  __/ |
 /_/   \_\/_/\_\___/|_| |_|____/ \___|_|    \_/ \___|_|
 Standard Edition                        Powered by AxonIQ

version: 4.6.7
2022-12-26 08:52:10.267  INFO 10496 --- [           main] io.axoniq.axonserver.AxonServer          : Starting AxonServer
 using Java 17.0.4.1 on DESKTOP-P2QERS8 with PID 10496 (C:\Users\Lahcen Lamkirich\Downloads\AxonServer\AxonServer-4.6.7\
axonserver.jar started by lahcen_lam in C:\Users\Lahcen Lamkirich\Downloads\AxonServer\AxonServer-4.6.7)
2022-12-26 08:52:10.272  INFO 10496 --- [           main] io.axoniq.axonserver.AxonServer          : No active profile s
et, falling back to 1 default profile: "default"
2022-12-26 08:52:15.838  INFO 10496 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized
with port(s): 8024 (http)
2022-12-26 08:52:16.527  INFO 10496 --- [           main] A.i.a.a.c.MessagingPlatformConfiguration : Configuration initi
alized with SSL DISABLED and access control DISABLED.
2022-12-26 08:52:21.645  INFO 10496 --- [           main] io.axoniq.axonserver.AxonServer          : Axon Server version
 4.6.7
```



SSL disabled

Authentication disabled

Configuration

Node Name	DESKTOP-P2QERS8
Host Name	DESKTOP-P2QERS8
Http Port	8024
GRPC Port	8124

Status

Last event token	4

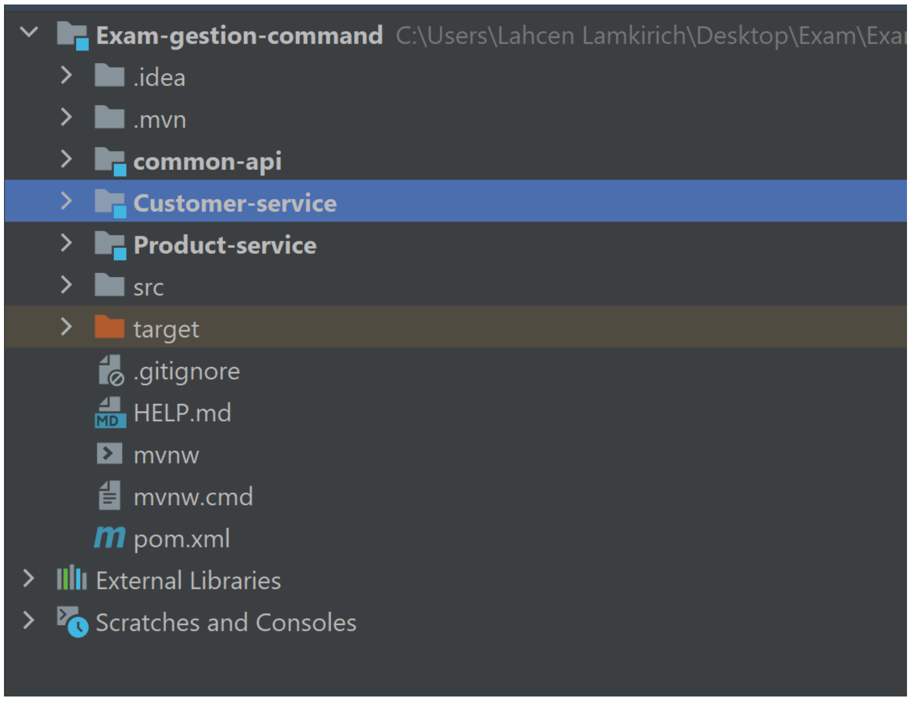
Activity in the last minute

Commands received / second	0
Queries received / second	0
Events stored / second	0
Snapshots stored / second	0

License

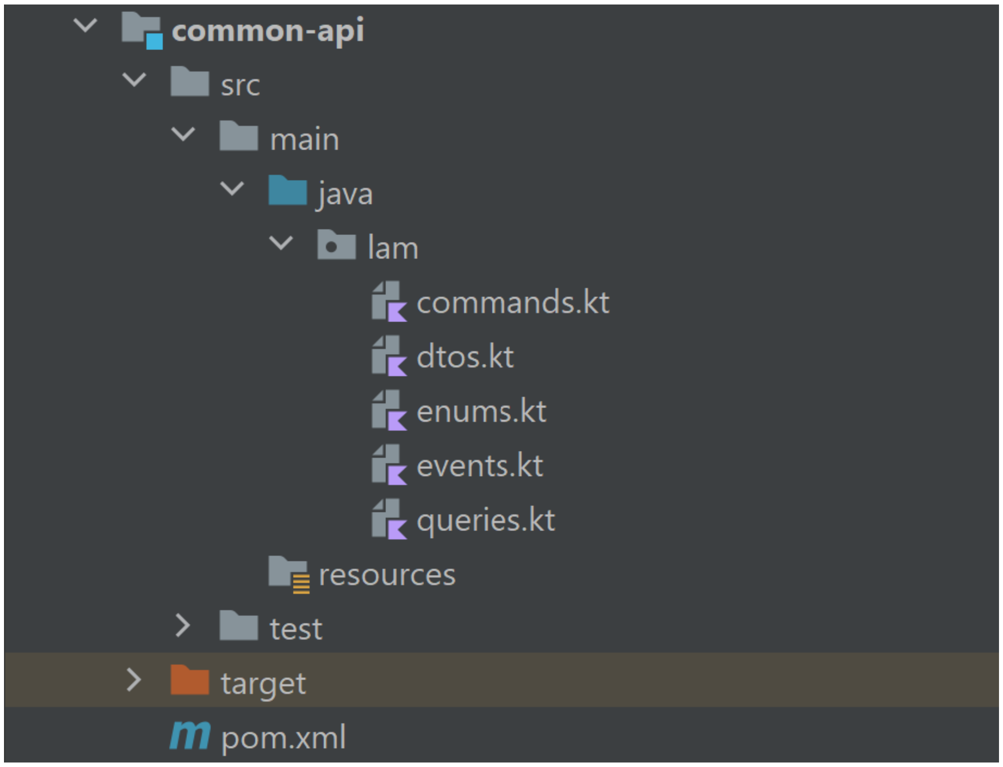
Edition	Standard Edition

⇨ **La présentation de la structure de projet**

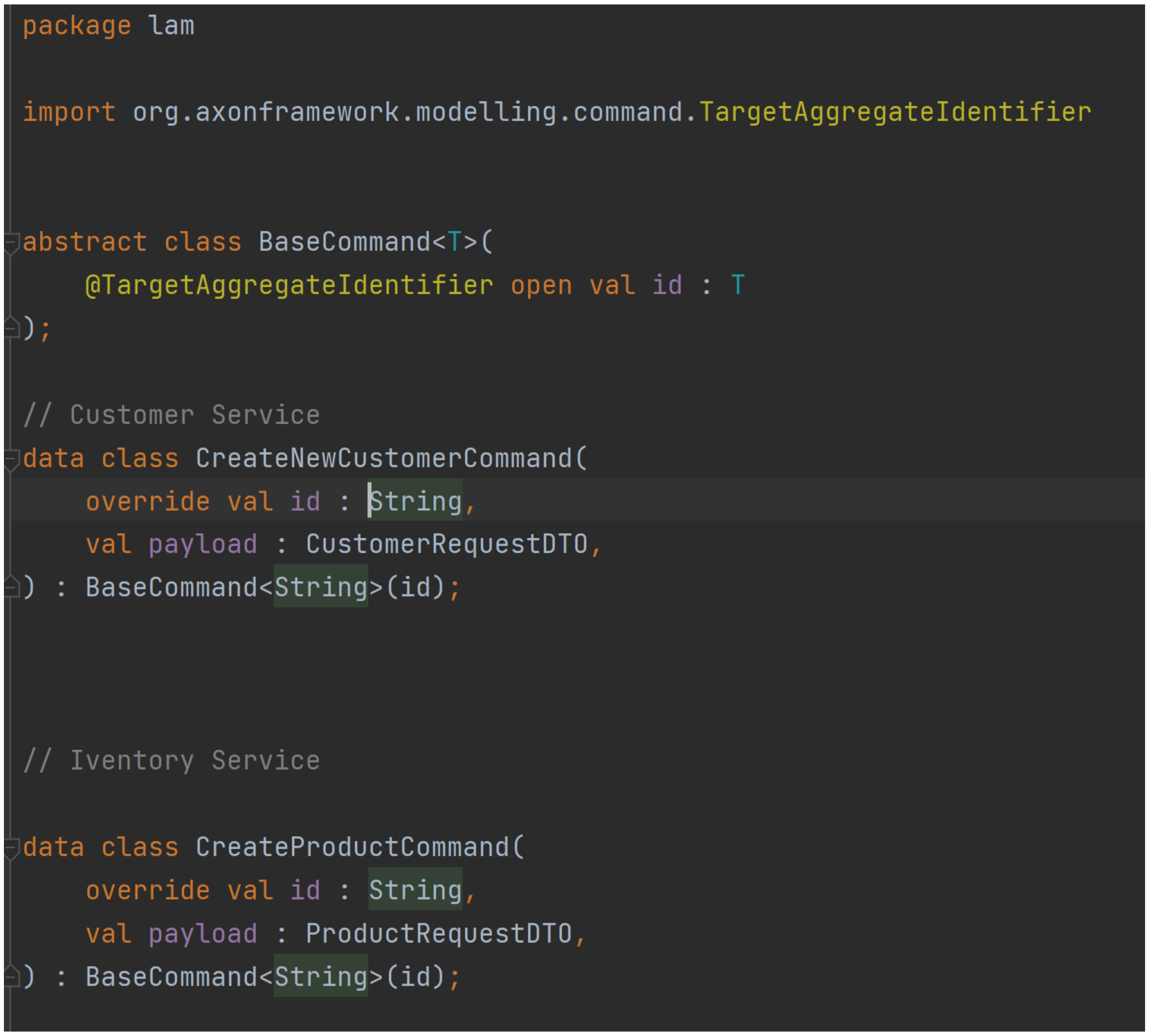


Exam-gestion-command C:\Users\Lahcen Lamkirich\Desktop\Exam\Exa
.idea
.mvn
common-api
Customer-service
Product-service
src
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
External Libraries
Scratches and Consoles

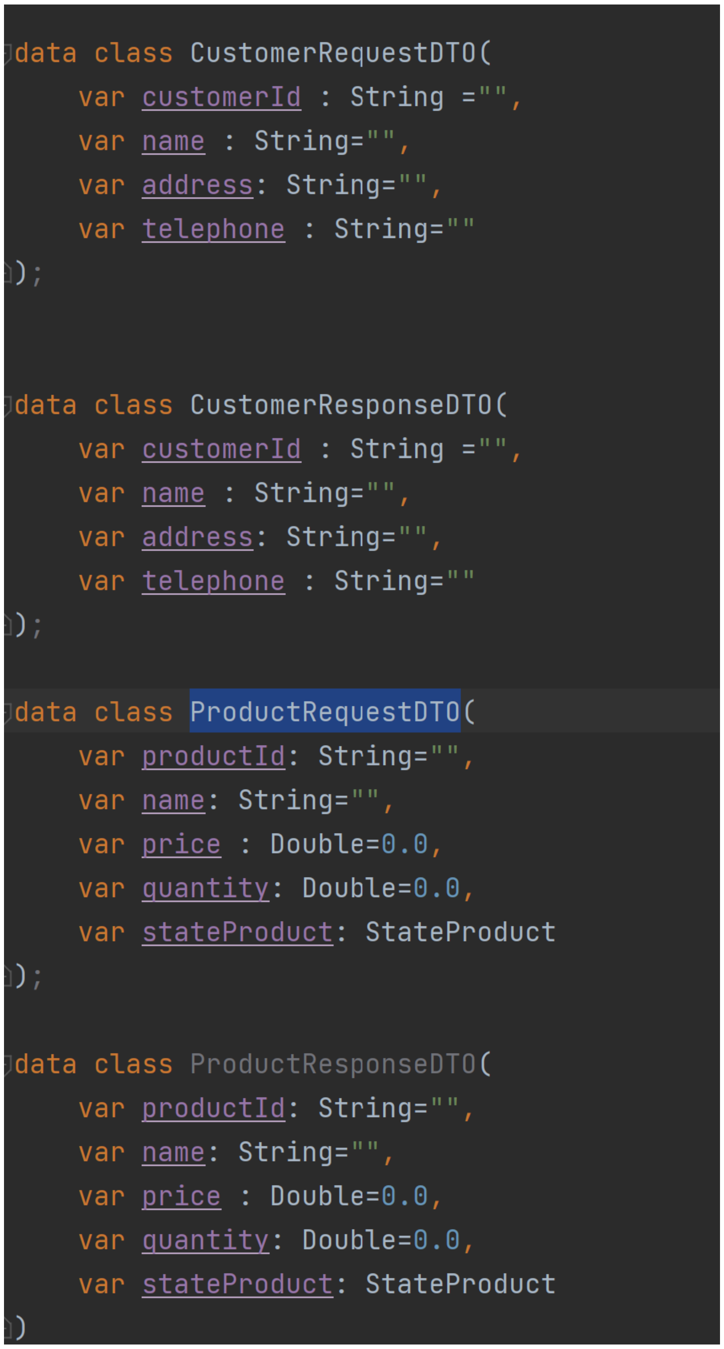
⇨ **La structure de Common API avec KOTLIN**

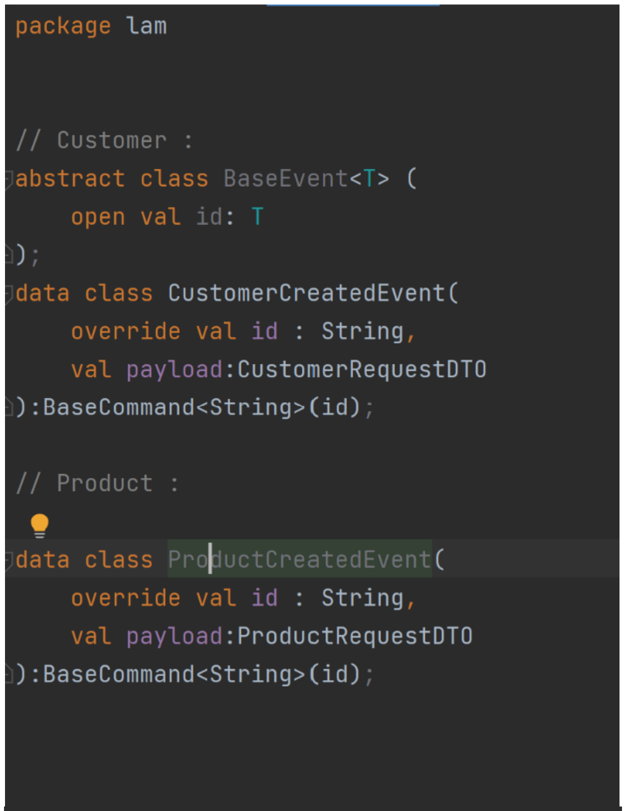


common-api
src
main
java
lam
commands.kt
dtos.kt
enums.kt
events.kt
queries.kt
resources
test
target
pom.xml

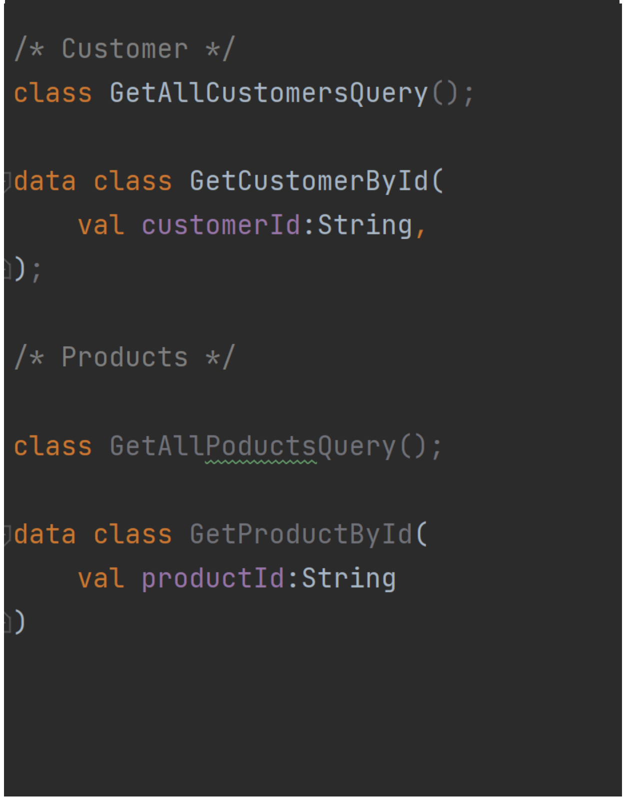


```kotlin
package lam

import org.axonframework.modelling.command.TargetAggregateIdentifier


abstract class BaseCommand<T>(
    @TargetAggregateIdentifier open val id : T
);

// Customer Service
data class CreateNewCustomerCommand(
    override val id : String,
    val payload : CustomerRequestDTO,
) : BaseCommand<String>(id);



// Iventory Service

data class CreateProductCommand(
    override val id : String,
    val payload : ProductRequestDTO,
) : BaseCommand<String>(id);
```

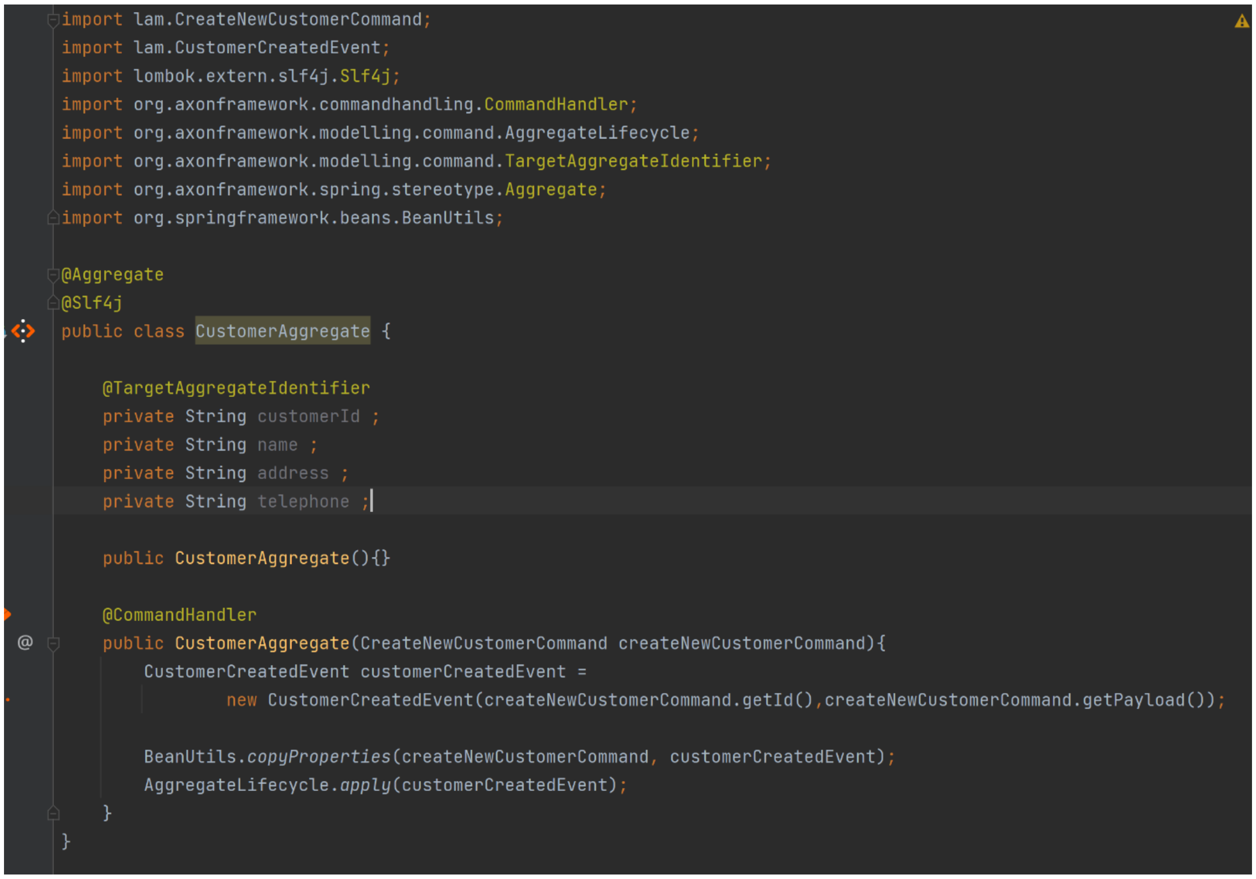

```kotlin
data class CustomerRequestDTO(
    var customerId : String ="",
    var name : String="",
    var address: String="",
    var telephone : String=""
);


data class CustomerResponseDTO(
    var customerId : String ="",
    var name : String="",
    var address: String="",
    var telephone : String=""
);

data class ProductRequestDTO(
    var productId: String="",
    var name: String="",
    var price : Double=0.0,
    var quantity: Double=0.0,
    var stateProduct: StateProduct
);

data class ProductResponseDTO(
    var productId: String="",
    var name: String="",
    var price : Double=0.0,
    var quantity: Double=0.0,
    var stateProduct: StateProduct
)
```

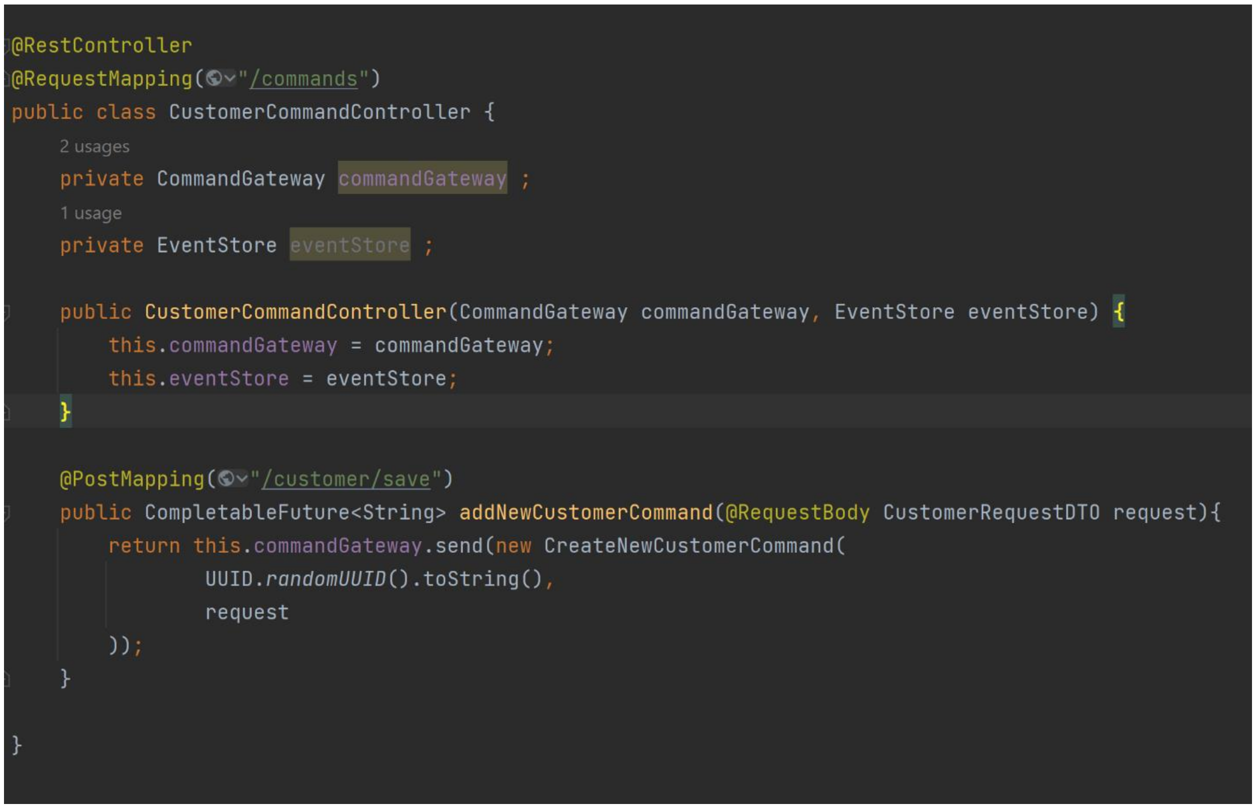


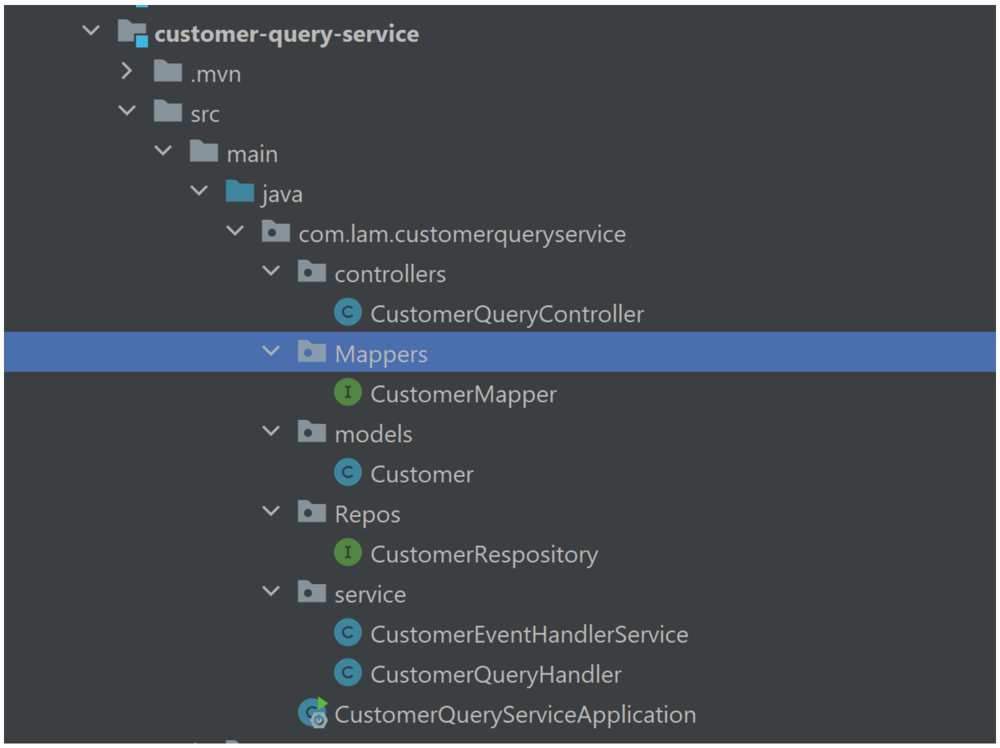
```kotlin
package lam


// Customer :
abstract class BaseEvent<T> (
    open val id: T
);
data class CustomerCreatedEvent(
    override val id : String,
    val payload:CustomerRequestDTO
):BaseCommand<String>(id);

// Product :

data class ProductCreatedEvent(
    override val id : String,
    val payload:ProductRequestDTO
):BaseCommand<String>(id);
```



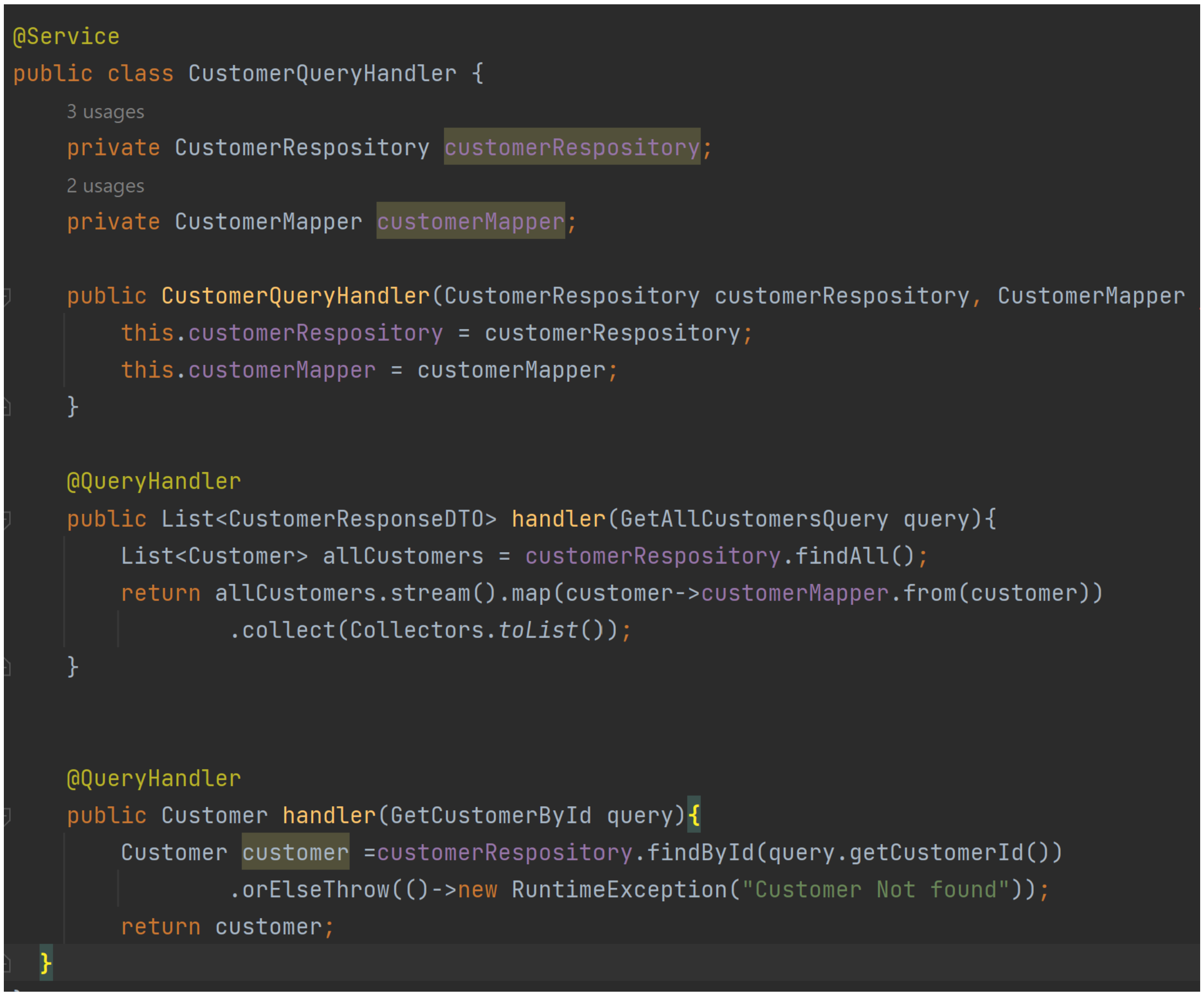
```kotlin
/* Customer */
class GetAllCustomersQuery();

data class GetCustomerById(
    val customerId:String,
);

/* Products */

class GetAllPoductsQuery();

data class GetProductById(
    val productId:String
)
```


```java
import lam.CreateNewCustomerCommand;
import lam.CustomerCreatedEvent;
import lombok.extern.slf4j.Slf4j;
import org.axonframework.commandhandling.CommandHandler;
import org.axonframework.modelling.command.AggregateLifecycle;
import org.axonframework.modelling.command.TargetAggregateIdentifier;
import org.axonframework.spring.stereotype.Aggregate;
import org.springframework.beans.BeanUtils;

@Aggregate
@Slf4j
public class CustomerAggregate {

    @TargetAggregateIdentifier
    private String customerId ;
    private String name ;
    private String address ;
    private String telephone ;

    public CustomerAggregate(){}

    @CommandHandler
    public CustomerAggregate(CreateNewCustomerCommand createNewCustomerCommand){
        CustomerCreatedEvent customerCreatedEvent =
                new CustomerCreatedEvent(createNewCustomerCommand.getId(),createNewCustomerCommand.getPayload());

        BeanUtils.copyProperties(createNewCustomerCommand, customerCreatedEvent);
        AggregateLifecycle.apply(customerCreatedEvent);
    }
}
```

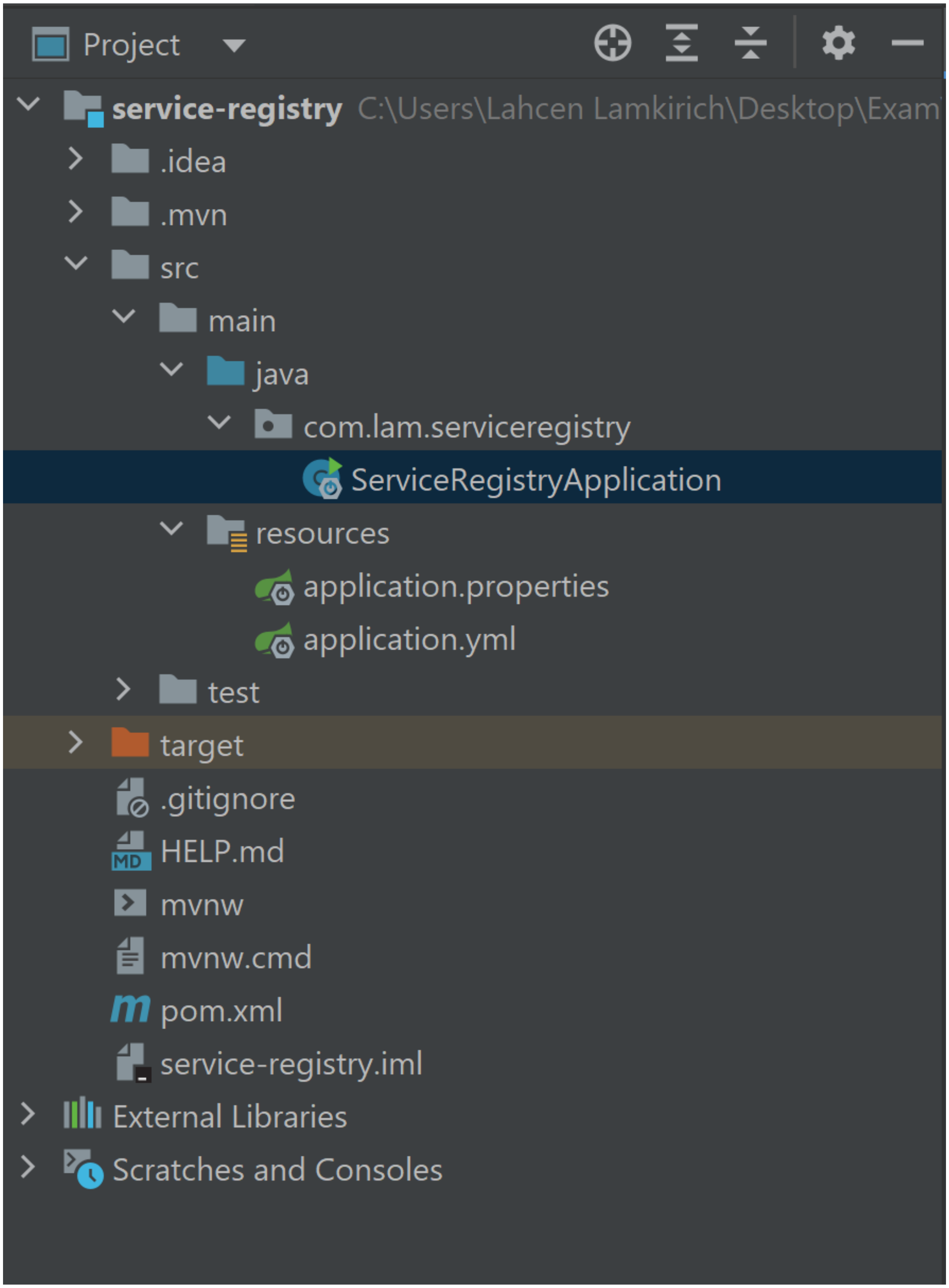

```java
@RestController
@RequestMapping("/commands")
public class CustomerCommandController {
    2 usages
    private CommandGateway commandGateway ;
    1 usage
    private EventStore eventStore ;

    public CustomerCommandController(CommandGateway commandGateway, EventStore eventStore) {
        this.commandGateway = commandGateway;
        this.eventStore = eventStore;
    }

    @PostMapping("/customer/save")
    public CompletableFuture<String> addNewCustomerCommand(@RequestBody CustomerRequestDTO request){
        return this.commandGateway.send(new CreateNewCustomerCommand(
                UUID.randomUUID().toString(),
                request
        ));
    }

}
```



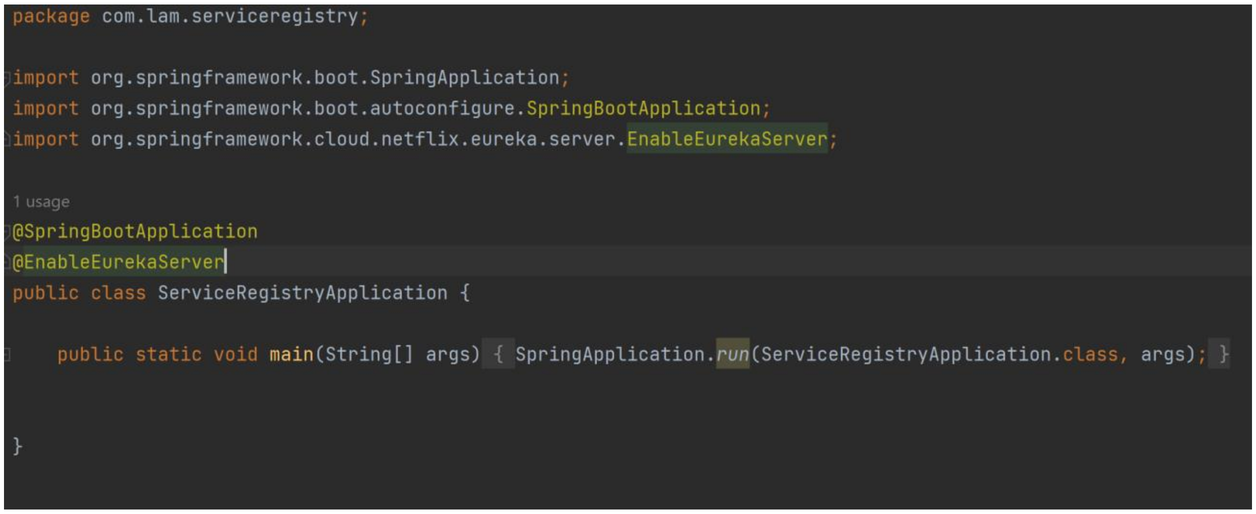
```
customer-query-service
  .mvn
  src
    main
      java
        com.lam.customerqueryservice
          controllers
            CustomerQueryController
          Mappers
            CustomerMapper
          models
            Customer
          Repos
            CustomerRespository
          service
            CustomerEventHandlerService
            CustomerQueryHandler
          CustomerQueryServiceApplication
```



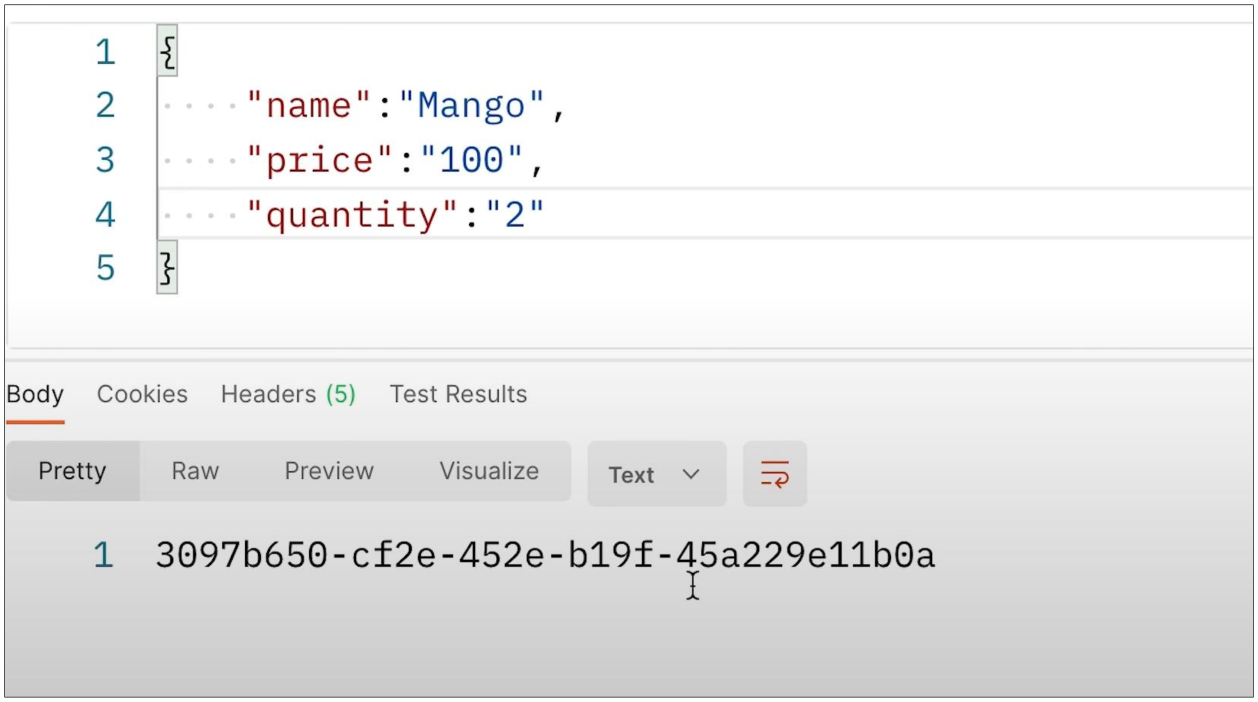
```
@Service
public class CustomerQueryHandler {
    3 usages
    private CustomerRespository customerRespository;
    2 usages
    private CustomerMapper customerMapper;

    public CustomerQueryHandler(CustomerRespository customerRespository, CustomerMapper
        this.customerRespository = customerRespository;
        this.customerMapper = customerMapper;
    }

    @QueryHandler
    public List<CustomerResponseDTO> handler(GetAllCustomersQuery query){
        List<Customer> allCustomers = customerRespository.findAll();
        return allCustomers.stream().map(customer->customerMapper.from(customer))
                .collect(Collectors.toList());
    }


    @QueryHandler
    public Customer handler(GetCustomerById query){
        Customer customer =customerRespository.findById(query.getCustomerId())
                .orElseThrow(()->new RuntimeException("Customer Not found"));
        return customer;
  }
```



```
service-registry  C:\Users\Lahcen Lamkirich\Desktop\Exam
  .idea
  .mvn
  src
    main
      java
        com.lam.serviceregistry
          ServiceRegistryApplication
      resources
        application.properties
        application.yml
    test
  target
  .gitignore
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml
  service-registry.iml
External Libraries
Scratches and Consoles
```



```java
package com.lam.serviceregistry;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

1 usage
@SpringBootApplication
@EnableEurekaServer
public class ServiceRegistryApplication {

    public static void main(String[] args) { SpringApplication.run(ServiceRegistryApplication.class, args); }


}
```



```json
{
    "name":"Mango",
    "price":"100",
    "quantity":"2"
}
```

```
3097b650-cf2e-452e-b19f-45a229e11b0a
```