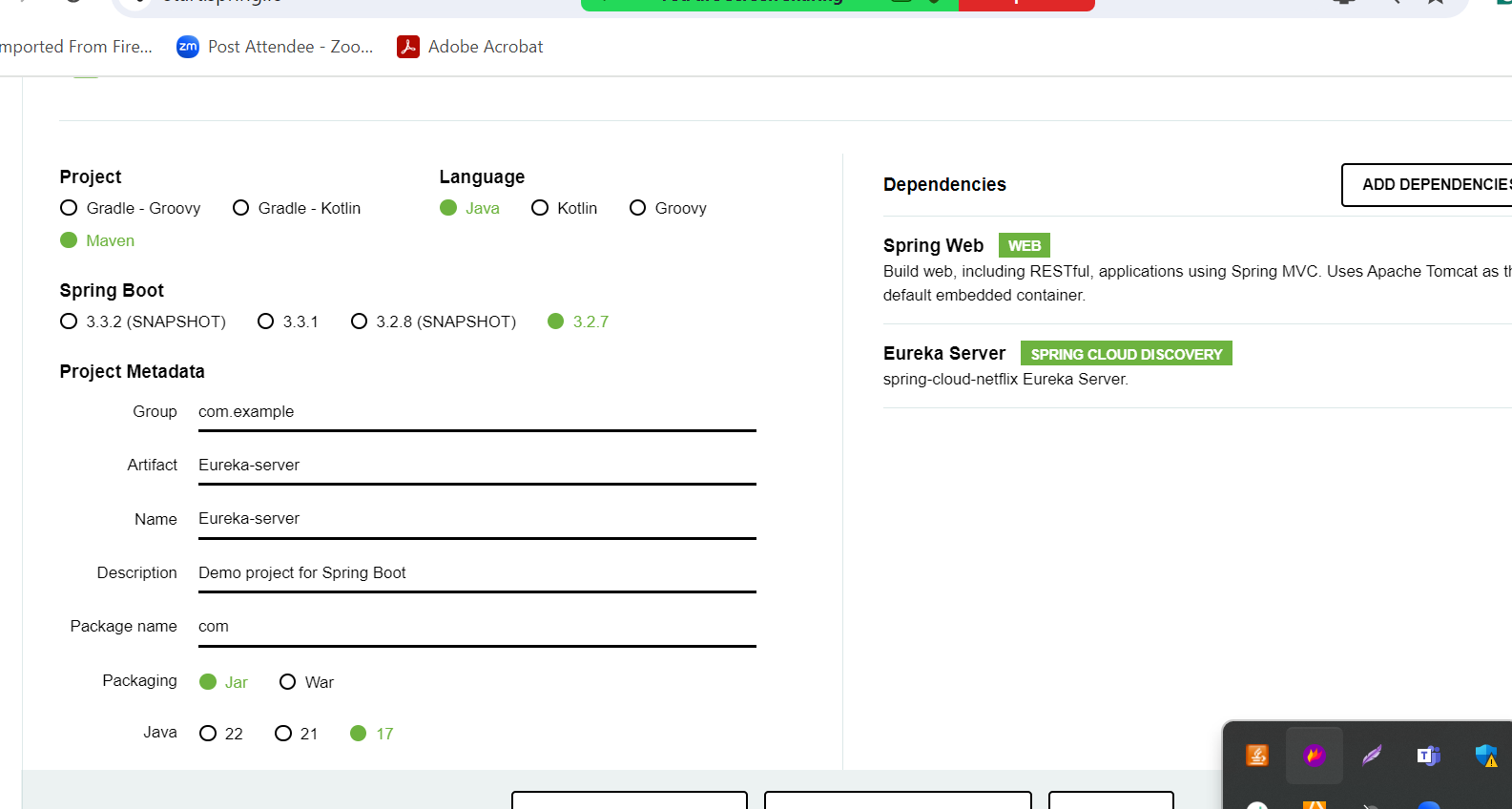
1st  project Eureka Server -🡪

Starter -🡪 Web Starter

Starter -🡪Eureka Server

Port number 8761 add two properties in application.properties file and run this project.



**application.properties file**

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

in main class @EnableEurekaServer we need to write and run the application.

2nd project micro service -🡪 book the cab

Starter 🡪Web Starter Controller

Stater 🡪JPA Starter

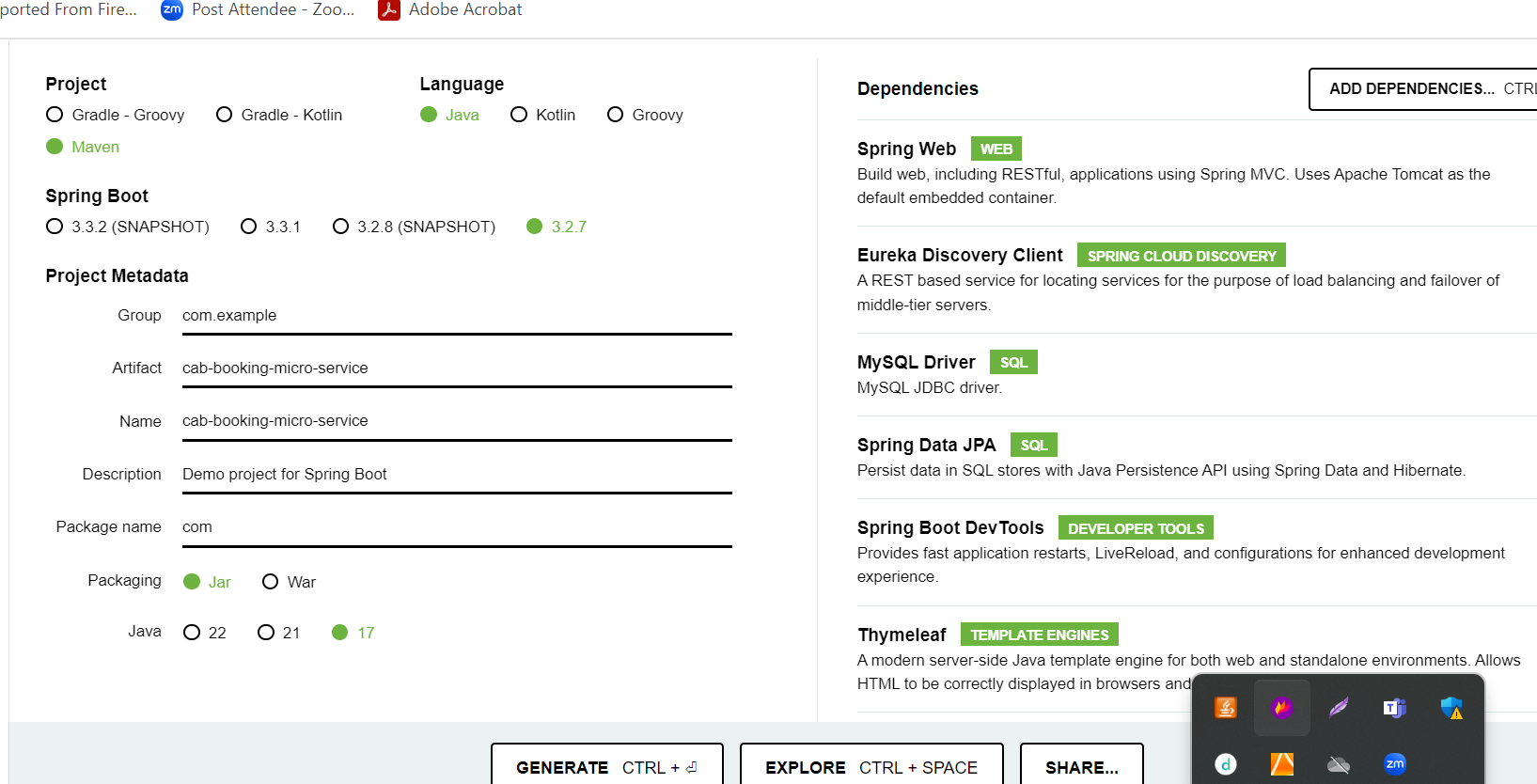
Dependency 🡪 MySQL

Starter 🡪Devtool

Starer 🡪Eureka Server

Starter -🡪Thymeleaf as View

Book the cab micro service project interact with cab fare service to get the amount.



application.properites

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/course3-db

spring.datasource.username=root

spring.datasource.password=root@123

spring.jpa.hibernate.ddl-auto=update

server.port=8181

@Entity

Entity class

@Id

bookingid(PK)

emailid (unique)

from

To

typeofcab

BookingRepository interface which extends JpaRepository

BookingService

Auto wired for BookingRepository as well as RestTemplate (which help use to call cab-fare service)

Bookthecab

Emailid (send data)

From (send data )

To (send data)

Typeofcab(send data)

Findthecarfare this method provide amount if cab available you can return cab book successfully else no cab available.

Findthecarfare using from, to, typeofcab with this information using restTemplate reference it will call car-fare microservice to get the amount.

Controller 🡪 normal @Controller annotation

First method to open index.page

Second method to book the cab

Third method to return cab fare amount

Write more if need.

Thymeleaf page

Index.html

Bookcab.html

Cabfare.html

Bookcab

Bookid(PK), emailidid (unique), from, to, typeofcab, amount etc.

3rd project micro service project to provide cab fare : provide the cab fare View as Rest API

Starter 🡪Web Starter RestController

Starter 🡪Jpa Starter

Dependency 🡪 MySQL

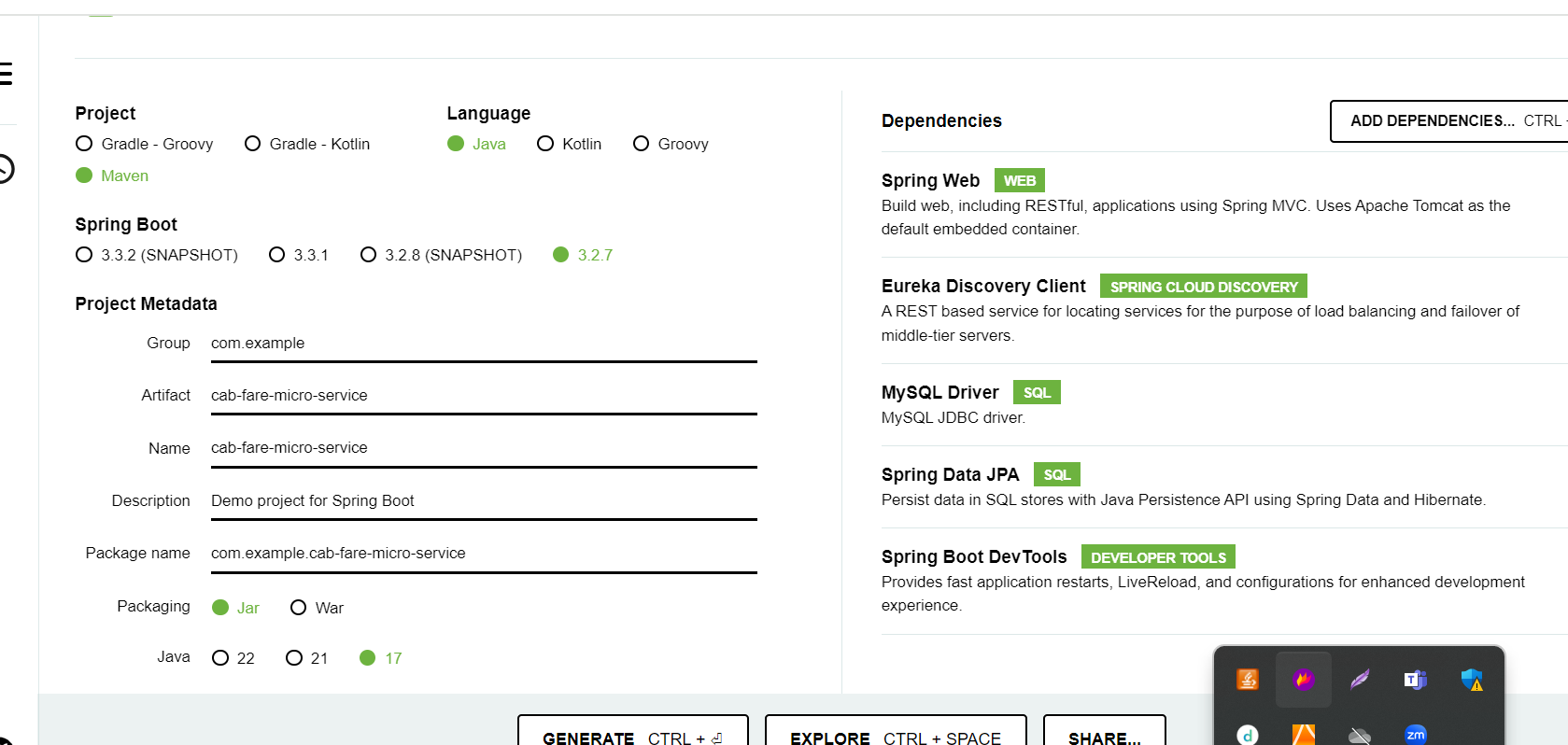
Starter 🡪devtool

Starter 🡪Eureka Client

8282

You need to create end point or rest api when we pass from, to, typeofcab you need to provide use cabfare. If cab is not available return the message else provide the fare or amount.

course3-db



application.properites

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/course3-db

spring.datasource.username=root

spring.datasource.password=root@123

spring.jpa.hibernate.ddl-auto=update

server.port=8282

@Entity

entity class :CabFake

@Id

cf(PK)

from

to

typeofcab

amount setter and getter methods

CabFareRepository which extends JpaRepository

Query(“Select cf.amount from CabFare cf where cf.to=:to and cf.from=:from and cf.typeofcab=:typeofcat”)

CabFareService

first method to insert carfare information (optional) or insert directly in db using insert query

Second method to find carfare using from, to and typeofcab using custom query part of repository interface.

CabFare rest controller

First method to store cabinfo using postmapping annotation ie post method (option)

Second method to get cab fare information using getmapping or postmapping (mandatory)

In main class

@SpringBootApplication with root package scan.

@EntityScan

@EnableJpaRepository

@EnableDiscovertyClient

cabfare table

cf(PK) from to typeofcab amount

1 A B innova 950

2 A B santro 650

3 B C innova 750