Micro Service : micro small service.



In monolithic service we need to write or create all modules using same language. Login module, dashboard module, customer module, product module, order module, payment module, manager module etc.

In real time every team or every person develop different modules. When we want to deploy we need to combine all these module and create jar or war file. We need to deploy all muddle as a package. If any issue any of the module the whole application can’t run.

In future if we need any changes in particular module we create update whole package once again after done the change in the module.

In micro service architecture we can deploy each module independently and those modules can be develop using same language or other language they can use same database or different database.

Python, spring boot, node with express js, asp.net etc

Mysql, oracle, mongo db, postgress etc

Login module create using spring boot with mysql database running on port number 8080

Product module create using spring boot with mysql database running on port number 8181

Interaction between one module to another using rest api.

customer module create using spring boot with mysql database running on port number 8282

to achieve micro service architecture using spring boot

spring boot provided two modules

spring cloud

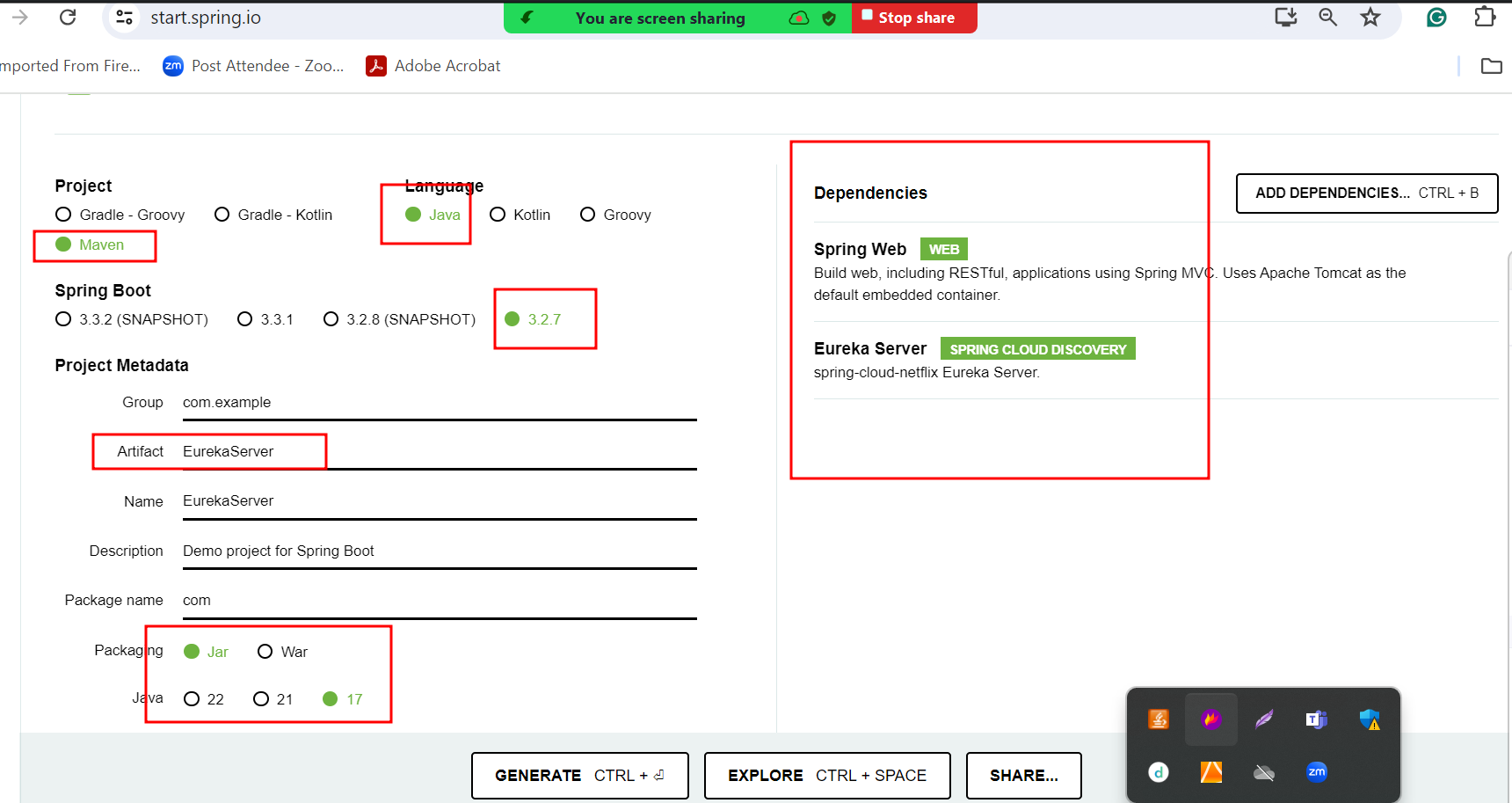
spring micro service

spring boot provided one of the open source server ie Eureka server. This server help us to deploy more then one micro service project.

1. Eureka Server project 🡪

Stater 🡪 web starter

Starter 🡪Eureka Server



If our project contains web starter by default tomcat start on port number 8080.

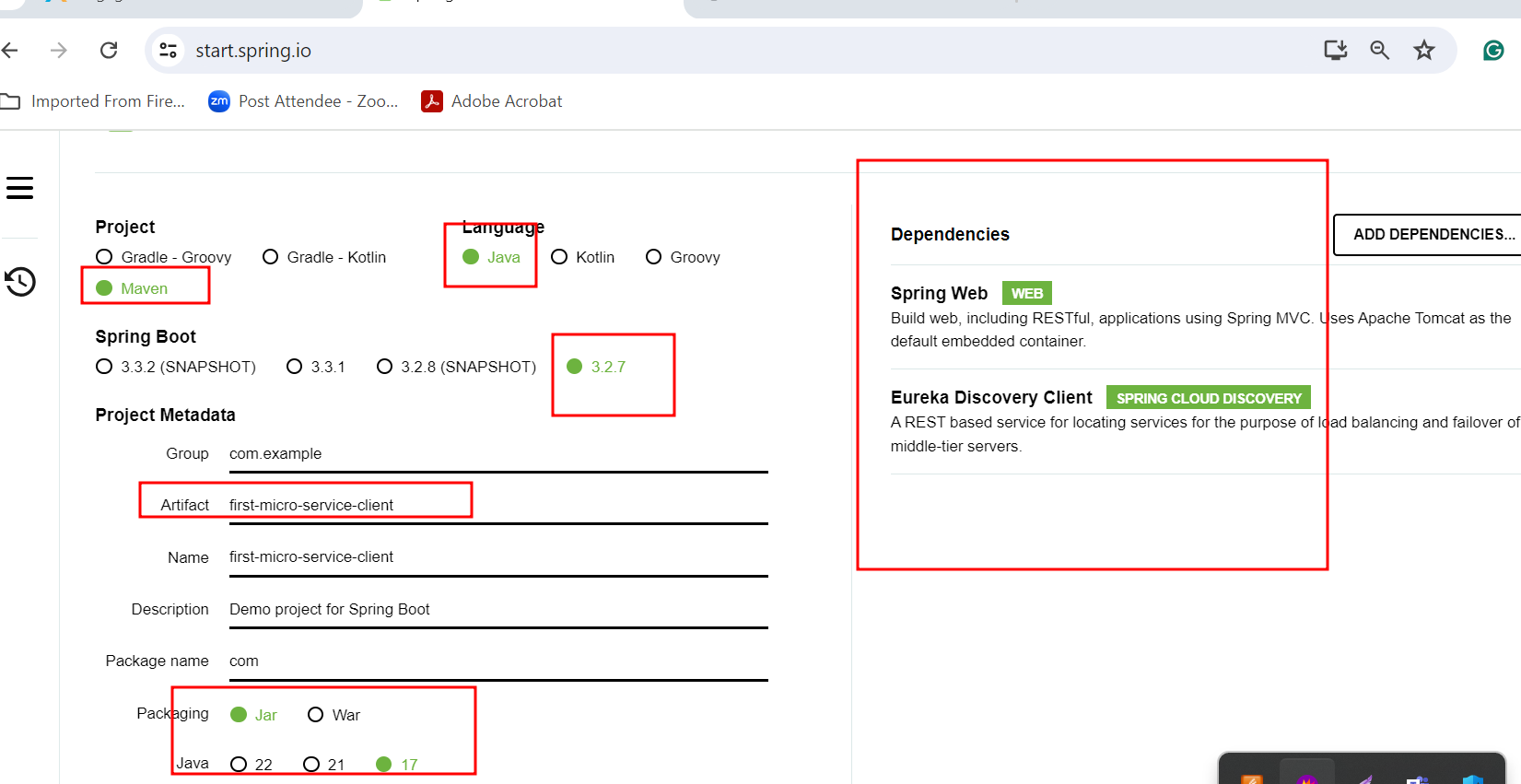
Whenever we deploy any micro service project each micro service application search by default port number for eureka server as 8761.

1. First-micro-service :

Starter 🡪web starter

Starter 🡪Eureka client

server.port =8181

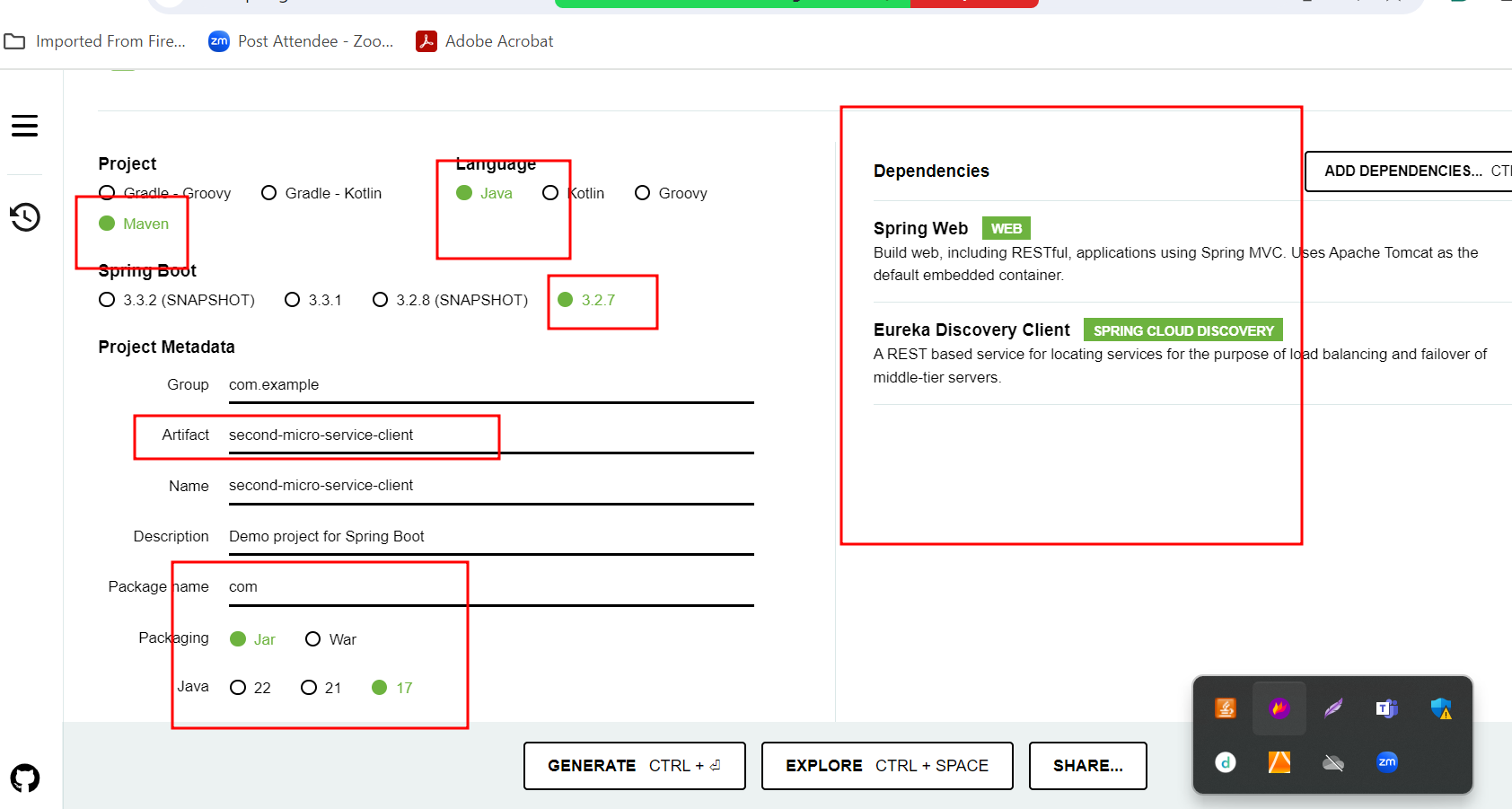


1. second-micro-service :

Starter 🡪web starter

Starter 🡪Eureka client

server.port =8282



1. account-micro-service :

Starter 🡪web starter

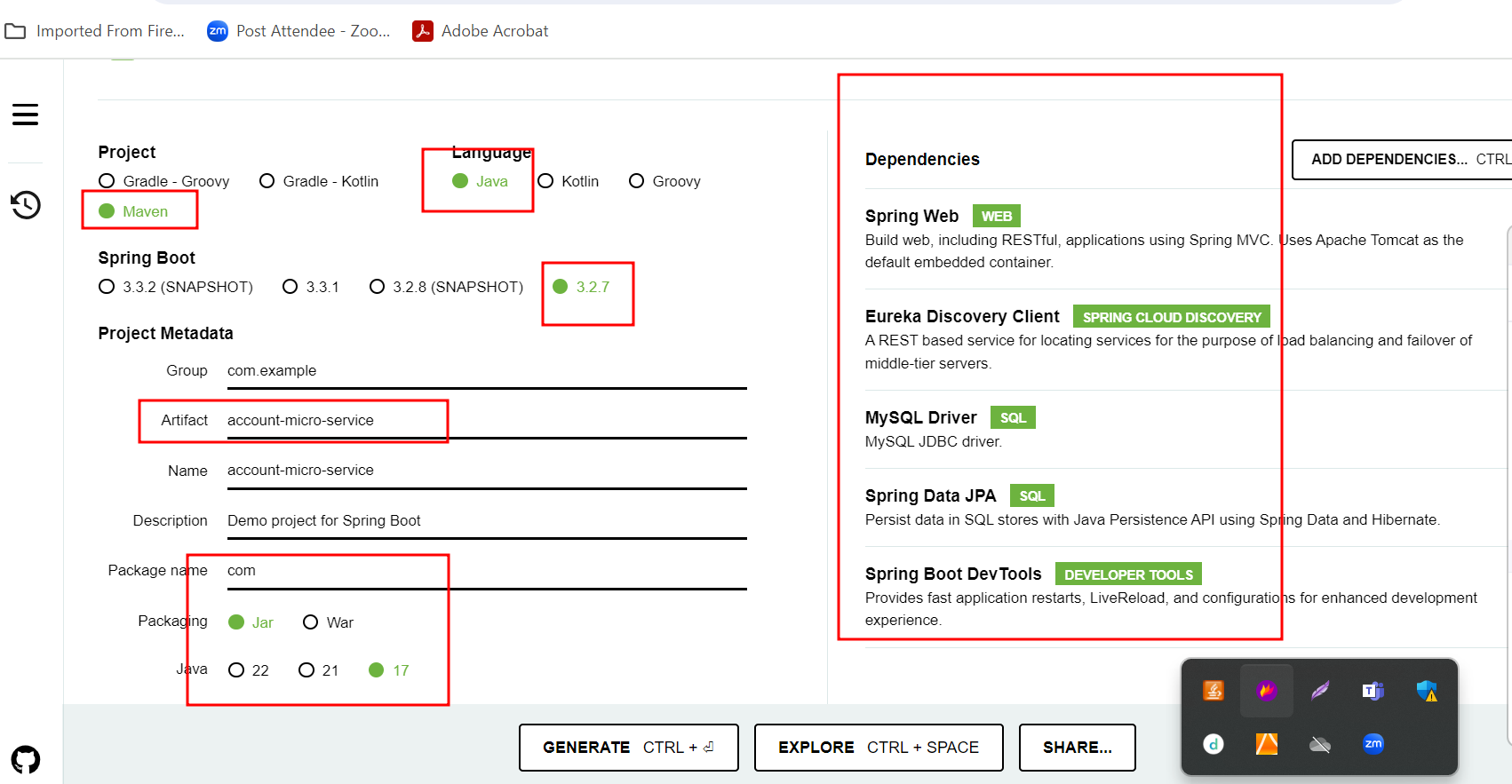
Starter 🡪jpa starter

Dependencies 🡪mysql

Starter 🡪devtool

Starter 🡪Eureka client

server.port =8383



mysql database. Db name as account-db

account table 🡪accno(accno pk, name, amount,emailid(unique))

create 🡪create account rest api post mapping

withdrawn 🡪 rest api put mapping

deposit 🡪 rest api put mapping

find account details 🡪 rest api get mapping

find accounnumber using emailid 🡪 respi get mapping

1. gpay-micro service

Starter 🡪web starter

Starter 🡪jpa starter

Dependencies 🡪mysql

Starter 🡪devtool

Starter 🡪Eureka client

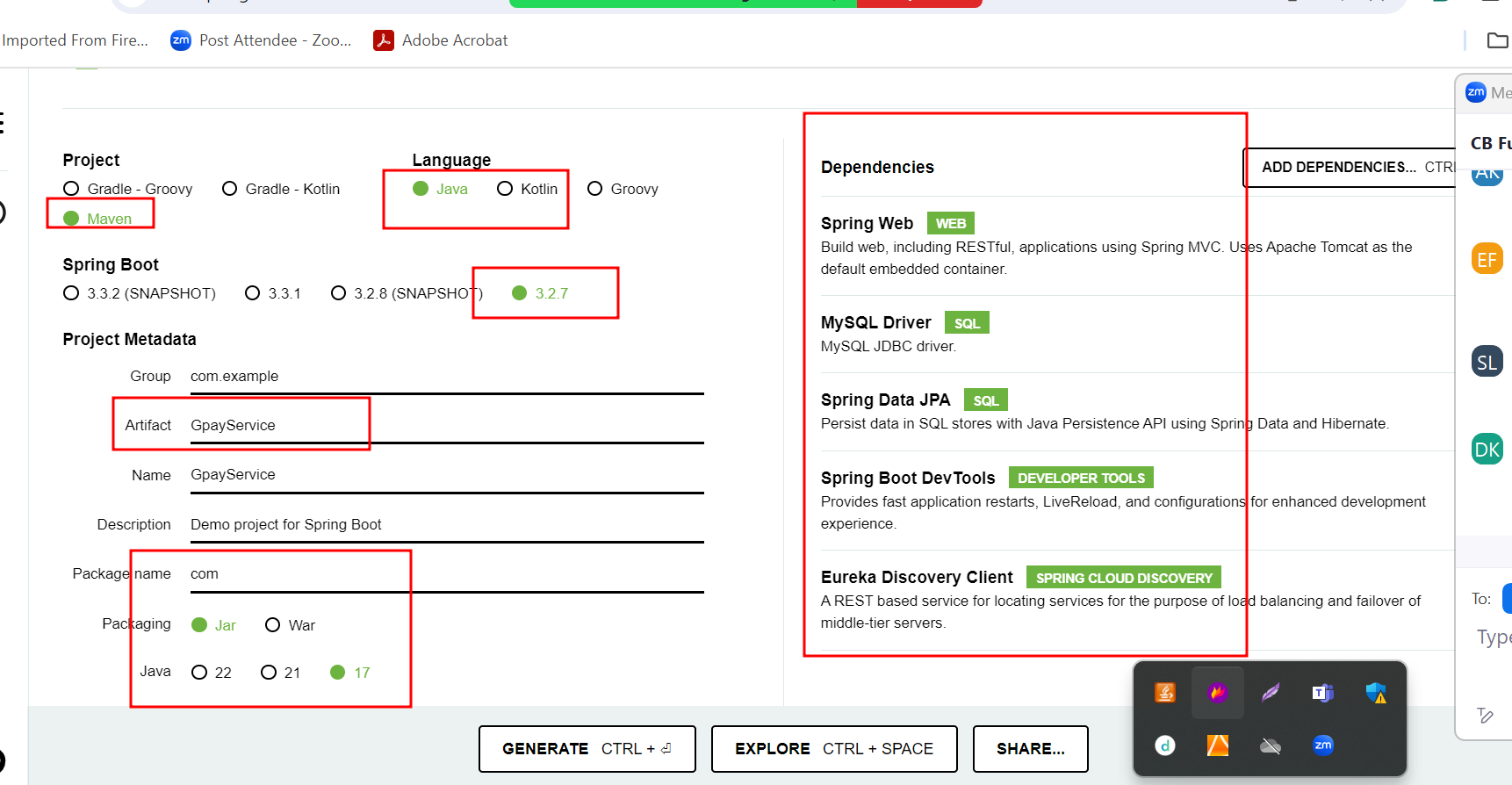
Gpay table (gid auto increment, emailid,accno )

Post mapping 🡪 create gpay account

Put mapping 🡪 payment -🡪emailid

Mysql database / oracle database db name as gpay-db

server.port =8484



@Bean : this annotation we can write on method level. That method is responsible to create the response of any type. Resource created by us but maintain by container.