Micro Service : Micro means very small service.



In Monolithic web service contains more than one modules like login modules(login controller, login service, login dao, login bean class. customer modules (customer controller, customer service, customer dao or repository and customer bean), account, transaction etc.

Dev1 Dev2 Dev3 Dev4

Login module Account module customer, Feedback module

We need to combine all those modules and need create single jar or war. Then that jar or war file we need to deploy or run on actual server ie production server environment.

CI And CD

Limitation of Monolithic web service.

1. All modules develop using same language.
2. One module depends upon another modules.
3. If we need to any changed in any one of the modules we need to do the changes and re-deploy whole application.

Login Module -🡪 Dev1

Using spring boot

LoginController

Login

LoginServie

LoginRepository

Database -🡪 mysql (mydb1)

They deploy the application.

8181

They communicate using rest api

Dashboard module dev2

Controller, service, repository and bean --🡪 database mysql (mydb2)

Then deploy the application

8282

If we want to implements application using Java as micro service

Spring framework provide two modules

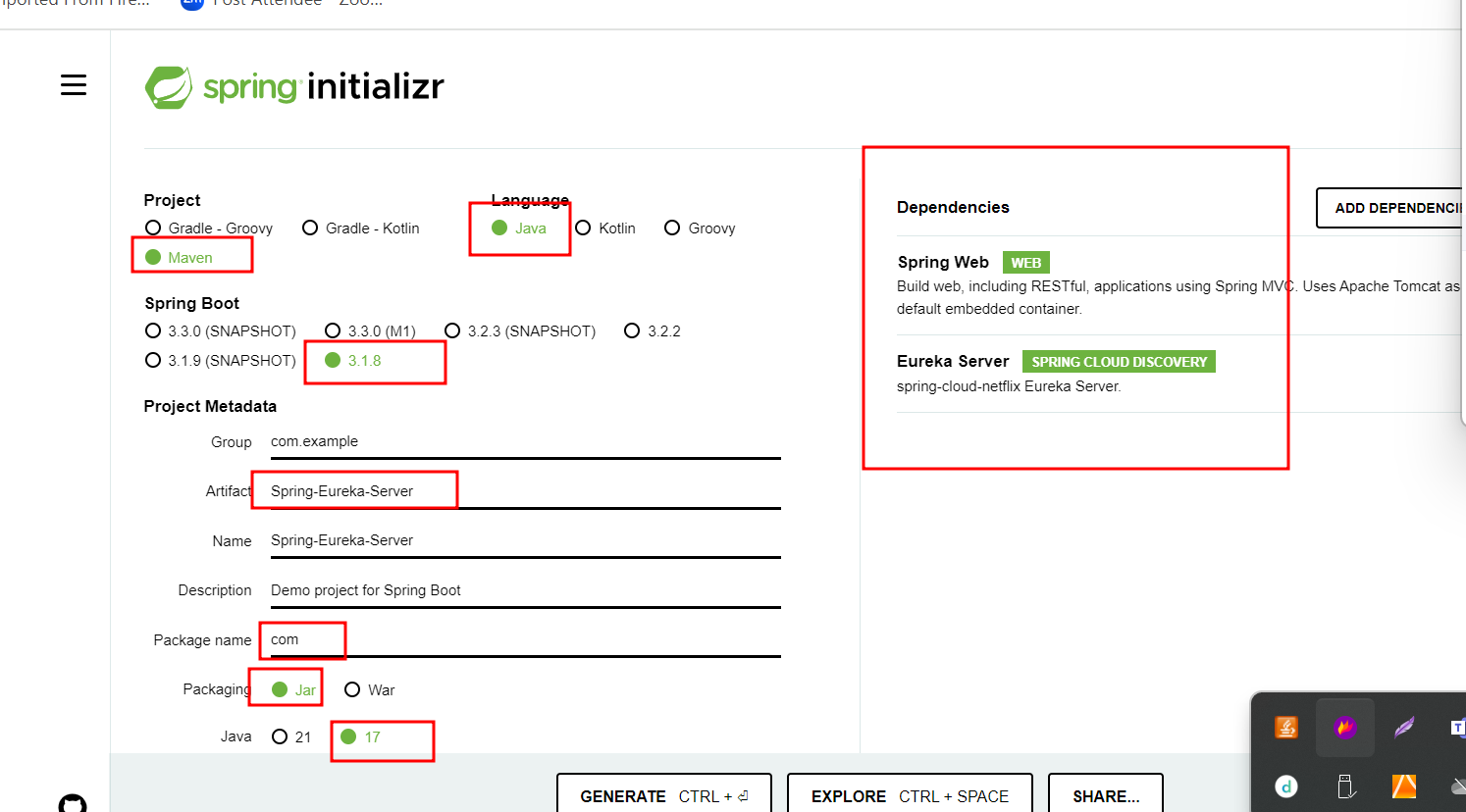
Spring cloud

Spring micro service

Spring Framework provided one of the web server ie Eureka Server. This server help us to deploy more than one micro service develop using spring framework.

Multi project application

1st project –(Eureka Server)🡪 contains web starter and Eureka Server starter which help to run the Eureka server.



In spring boot whenever we add web starter by default tomcat run on default port number 8080. Eureka server also run on default port number 8080.

If we do changes in eureka server project as 8761. The by default all micro service deploy on same port number.

But whenever we are going to deploy more than one micro service each service check default port number of eureka ie 8761 but if eureka server running on 8080 port number we need to mention in each micro service project eureka server running on 8080.

application.properties

server.port=8761

eureka.client.register-with-eureka=false

eureka.client.fetch-registry=false

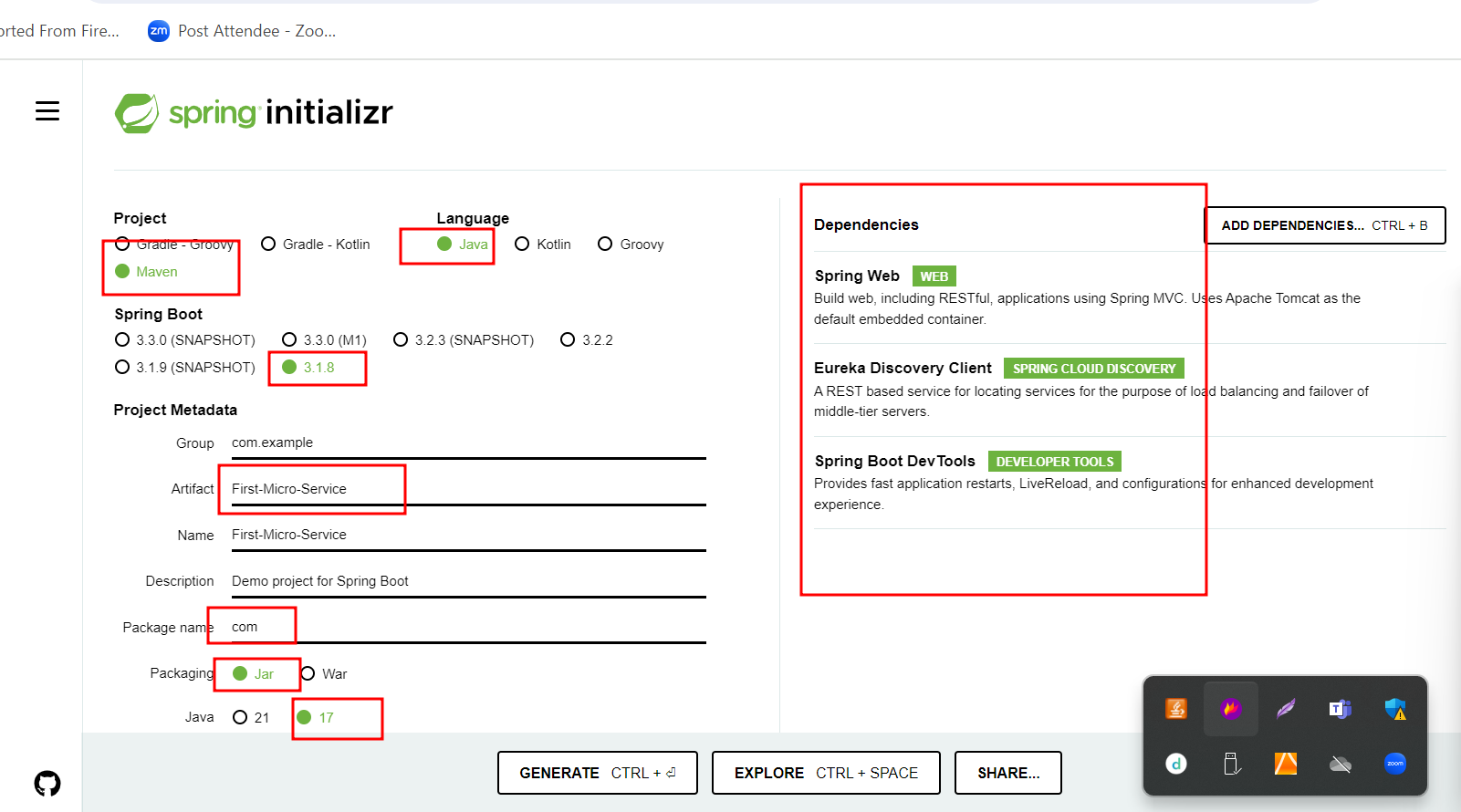
Eureka server project is a server application. This project is not providing any service it help us to deploy more than one micro service project.

@EnableEurekaServer

This annotation is use to enable eureka server features.

2nd project –(Eureka Client) 🡪 contains web starter and Eureka Client starter which help to deploy that project in Eureka Server.

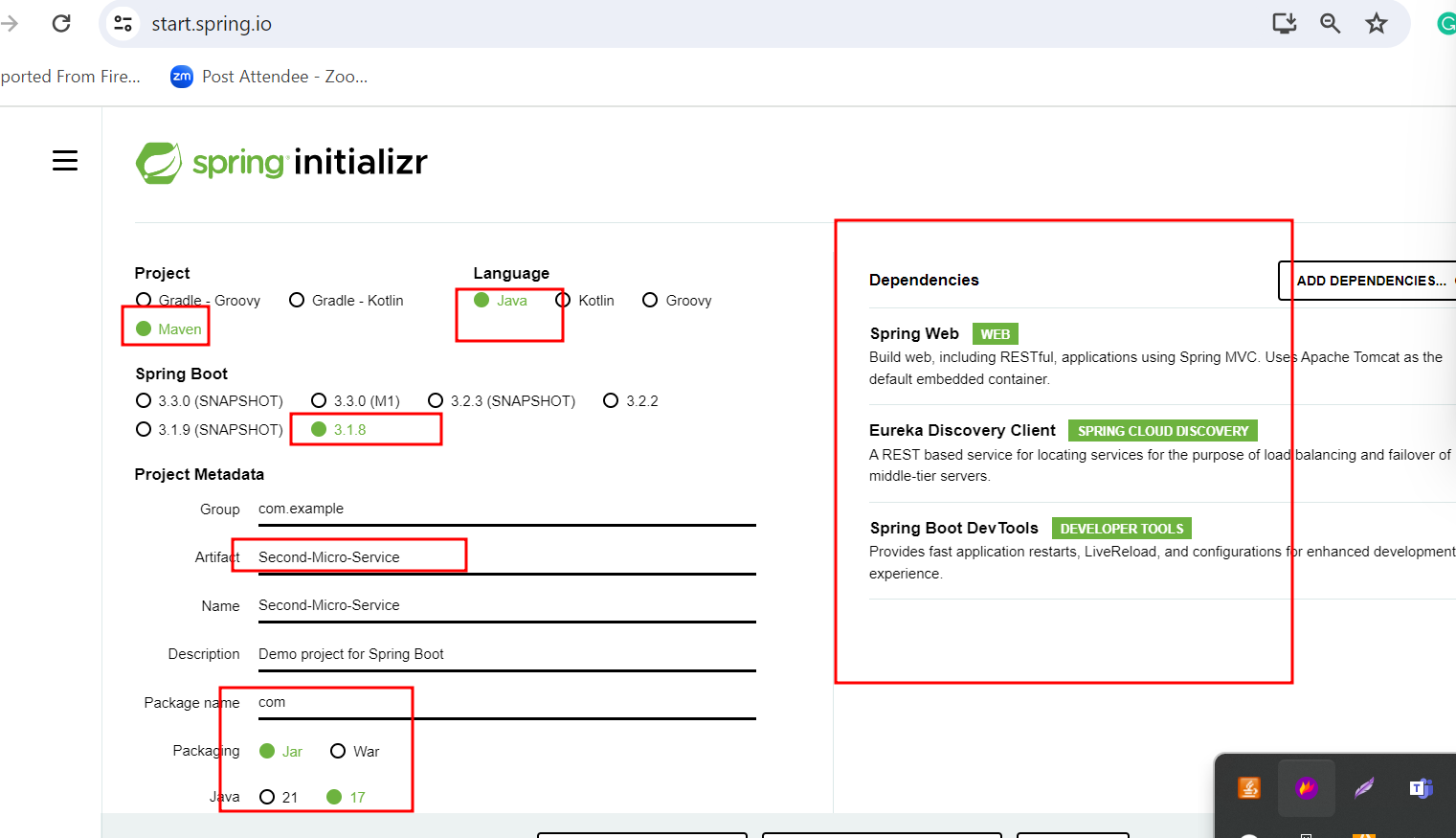
Micro service project. (this project only provide simple rest full message as welcome to spring boot micro service).



Port number 8080 (default port number).

3rd project –(Eureka Client) 🡪 contains web starter and Eureka Client starter which help to deploy that project in Eureka Server.

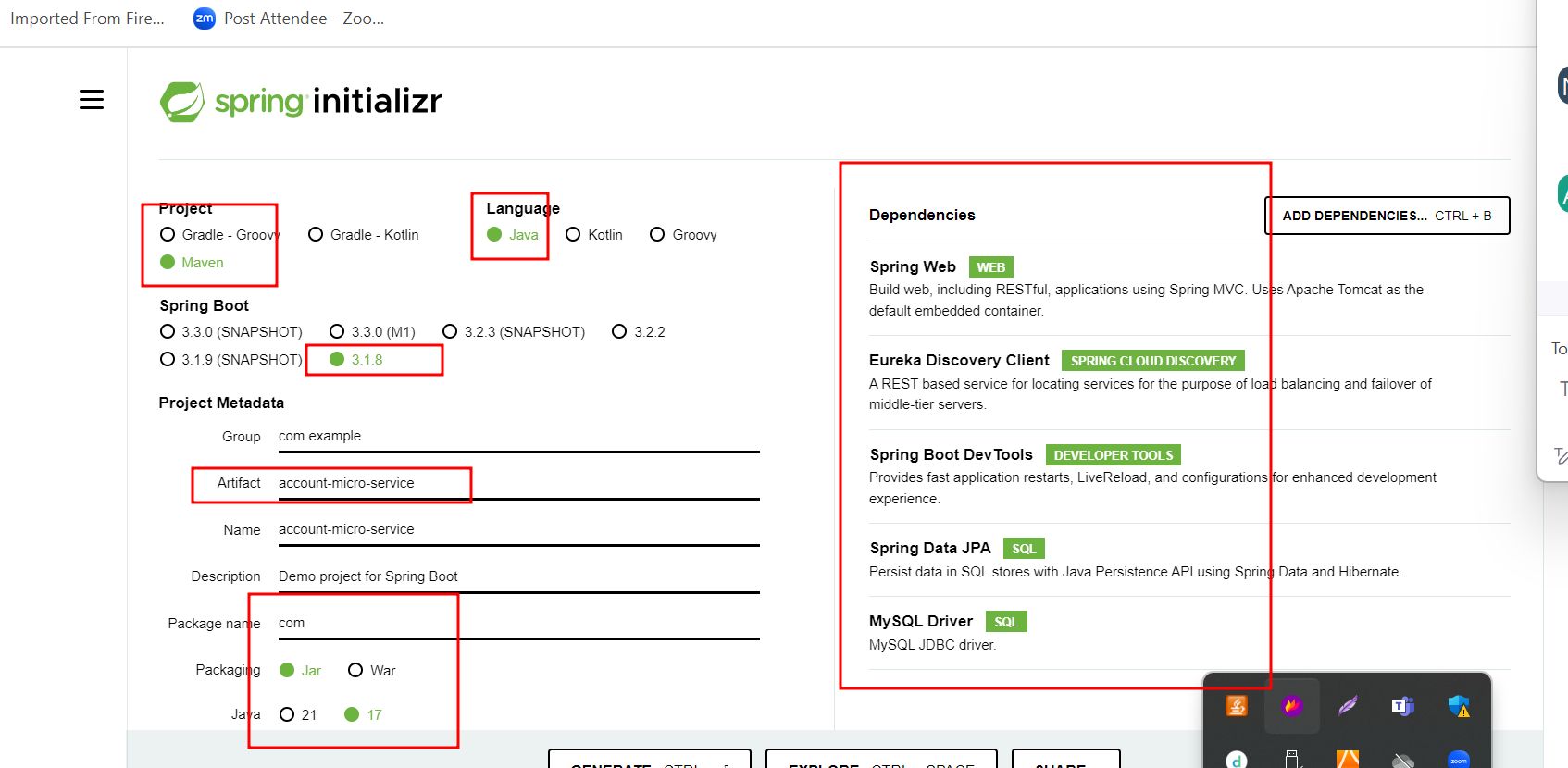
Micro service project. (this project only provide simple rest full message as welcome to spring boot micro service).



Port number 8080 (default port number).

4th project –(Eureka Client) 🡪 contains web starter and Eureka Client starter , jpa starter, mysql connector and devtools etc which help to deploy that project in Eureka Server. This project interact with database.

Account-micro-service



Port 8282

@RequestMapping Vs @GetMappng, @PostMapping, @DeleteMapping, @PutMapping

@ReqeustMapping annotation we can use on class level to check base path as well as method level with method attribute get get, post, put and delete etc. this is generic annotation

@Getmapping : this annotation we need to use on method level and it is consider as Get method.

Account -🡪 accno(PK), name ,emalid(unique), amount

8282

To communicate two micro service

GPay or Paypal -🡪 micro service ---🡪 develop using spring boot in different database.

GPay -🡪 gpid (PK), emaliid etc

8383

Using 8383 port number with emailed we can find account details.

GPay micro service

Web starter

Eureka Client

Jpa starter

Mysql connector

Devtool

