Day 10 : 25 Sep 2024



To create micro service architecture application using spring boot. Spring boot provided spring cloud modules. They provided us Server ie Eureka Server. Eureka Server is a type of registry or server which allow us to deploy more than one micro service projects. Eureka Server handle all configuration details about service, provide load balance feature, it check health of the application etc.

To implicit micro service architecture we need minimum two projects

1 Eureka Server

2. Eureka Client 1 8181

3. Eureka Client 2 8282

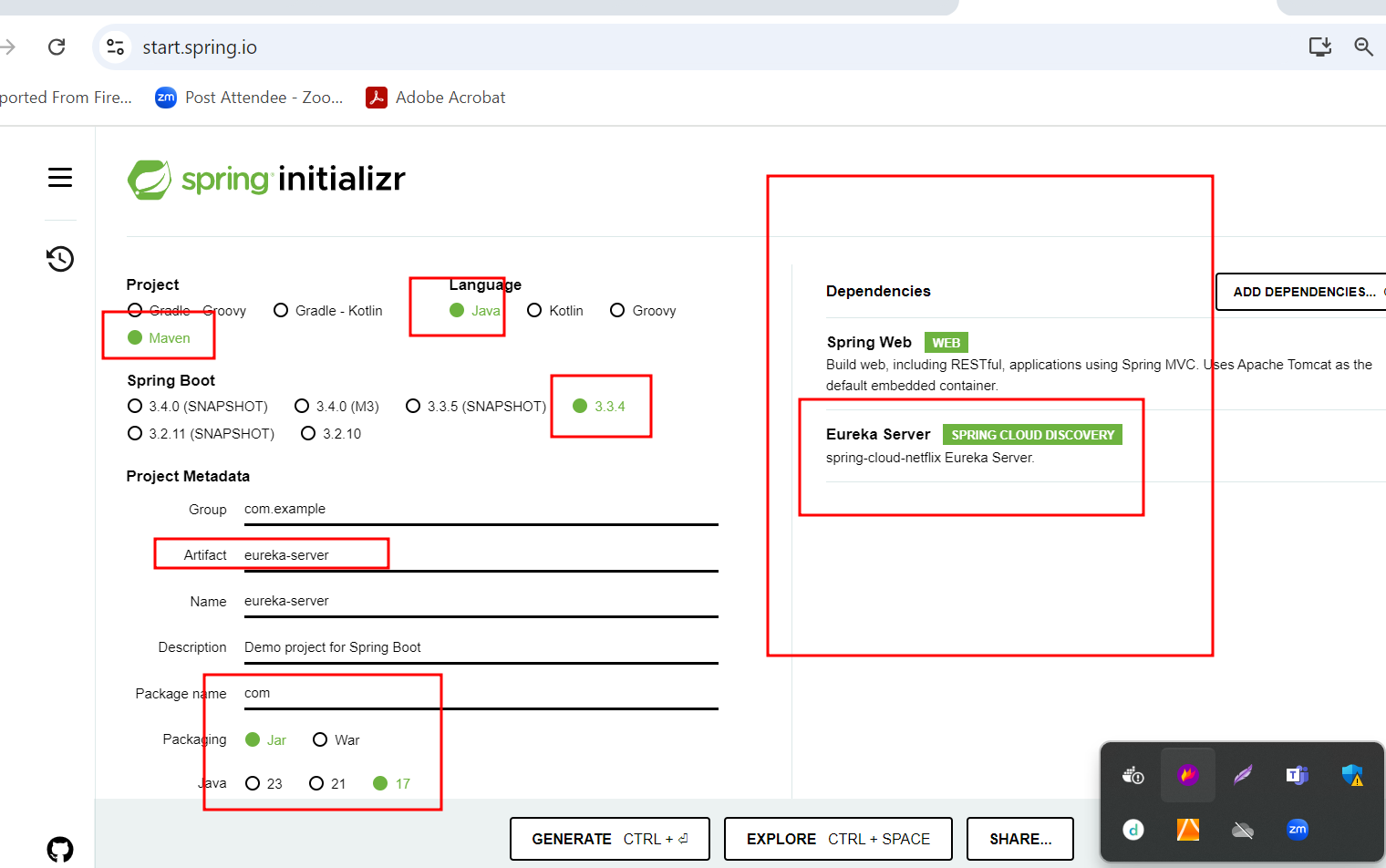
4. Eureka Client 3 8383

5. Eureka Client 4 8484

Eureka Server :

Web Starter

Eureka Starter



By default eureka server run on default port number 8761 port.

But we added web starter so eureka server run on port number 8080.

But in every micro service project we need mention eureka server port number. Because every micro service with eureka client application search eureka on port number 8761.

1. Account Micro Service 8383

Web Starter -🡪 end point

Eureka Client --🡪 help us to deploy project on Eureka Server

Jpa starter -🡪 orm features

My sql connector -🡪 mysql database

Dev tools --🡪 refresh the project

Lombok -🡪 to avoid setter and getter methods

Database 🡪 mysql

Post method : create account (accno(PK), name,amount,emailid(unique)) etc

Put method : withdrawn

Put method : deposit

Get method : using acno get amount

Get method : using emailid get the accno

1. Gpay, Paypal, Paytm etc 8484

Web starter 🡪 end point

Eureka Client --🡪 help us to deploy project on Eureka Server

Jpa starter -🡪 orm features

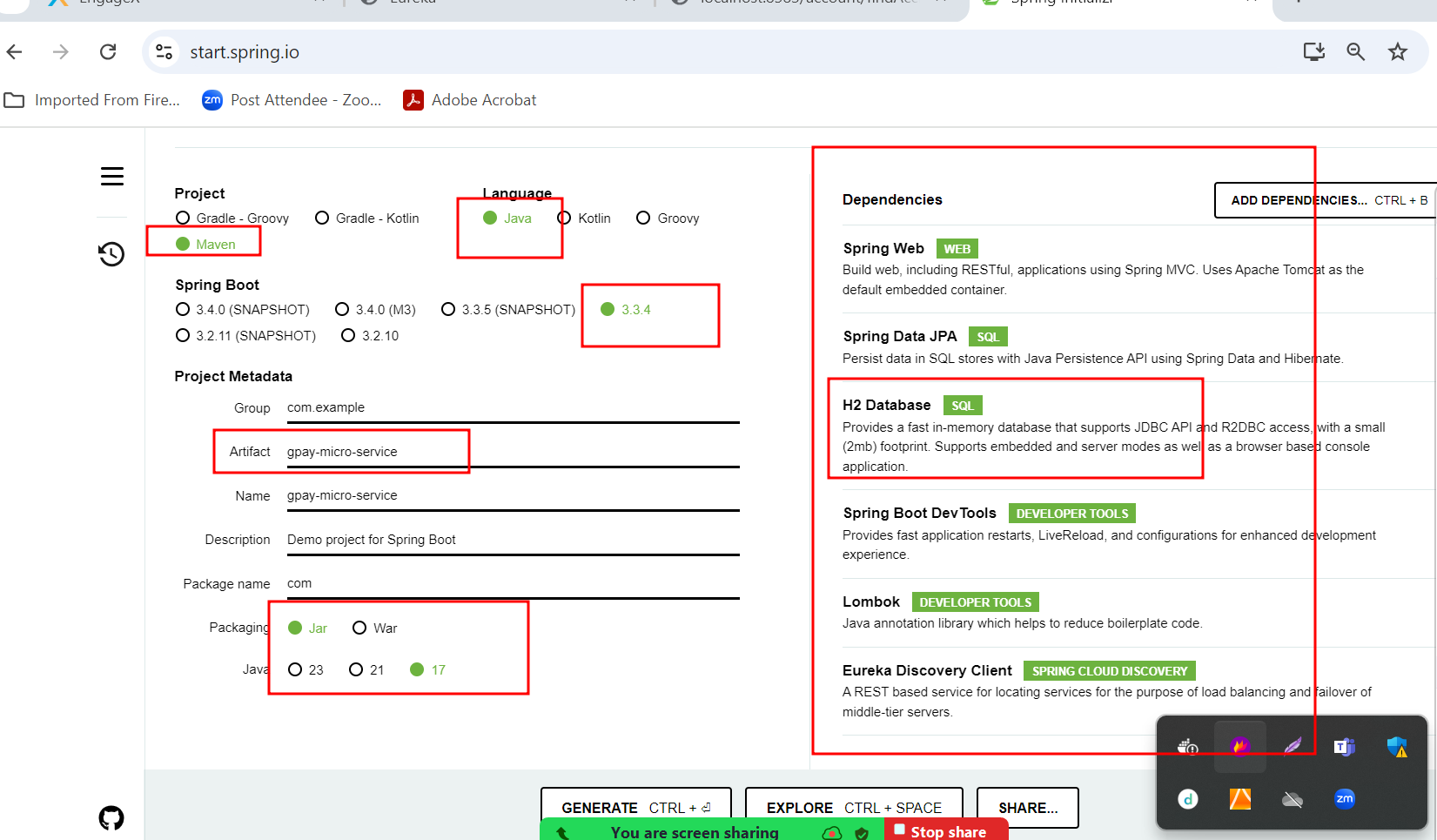
H2 connector 🡪 h2 in memory database.

Dev tools --🡪 refresh the project

Lombok -🡪 to avoid setter and getter methods

Database 🡪 h2

Post method -🡪while creating gpay,(gpaid auto increment, accno) from client side we ask emailid using emaild we interact with account micro service to get the account details. If present we create gpay account



In Spring boot we can communicate to another rest api or micro service using

One of the way RestTemplate