Day 9: 5 Oct 2024

Micro Service : micro means small. Small service responsible to do simple or small task. Using micro service we can create more than one service develop using same language or different language with same db or different db and deploy independently with different port number in same type of server or different type of server.

Before Micro Service

Login module : LoginController, Login entity , LoginService, LoginRepostory and database.

Customer module

Product module

Order module

Payment module

Etc

After develop all the modules we need to merge those module or integrate those module in shared repository ie github.

After merge all team code we need to re-build ie create jar or war file and test it and deploy on production environment.

1. One module or team depends upon another team module or code.
2. If we want to do any changes, after changes on that particular module we need to re compile and rebuild and deploy
3. If any module get any issue the whole application can’t work.
4. All modules develop using same language with same database.



To implements this micro service using spring boot.

Spring boot provided two modules

Spring cloud

Spring micro service

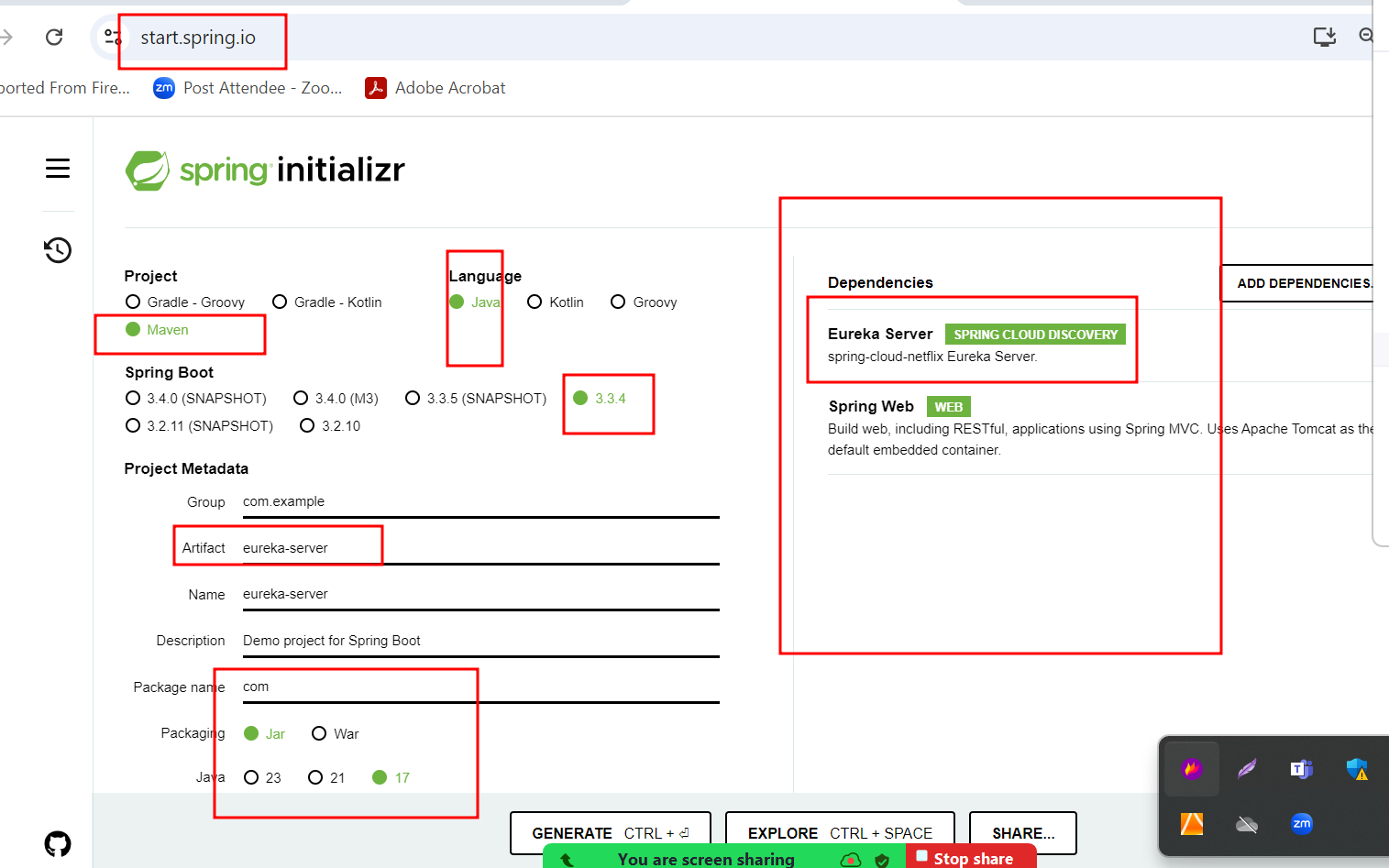
Spring cloud provided one of the pre defined open source server ie EurekaServer part of Netflix. This server help us to deploy more than one micro service created using same language or different language.

This server keep the track about health of the service.

1s project creating Eureka Server

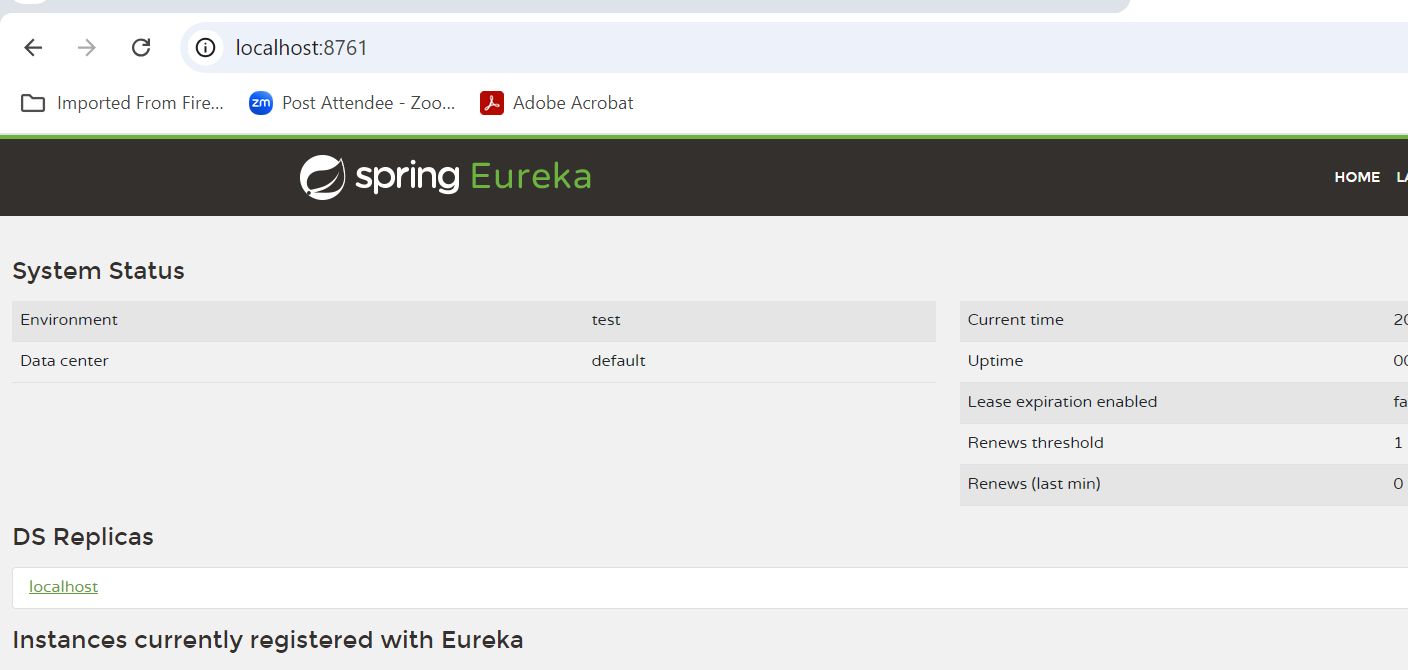
Starter -🡪 Eureka Server

Starter 🡪 Web Starter



Eureka Server default port number is 8761. Because we added web starter which contains tomcat so Eureka server by default run on port number 8080.

But whenever we run more than one micro service project in application.properties file we need to mention eureka server running on port number 8080. Because each micro service search eureka server by default on same machine ie localhost with port number 8761.



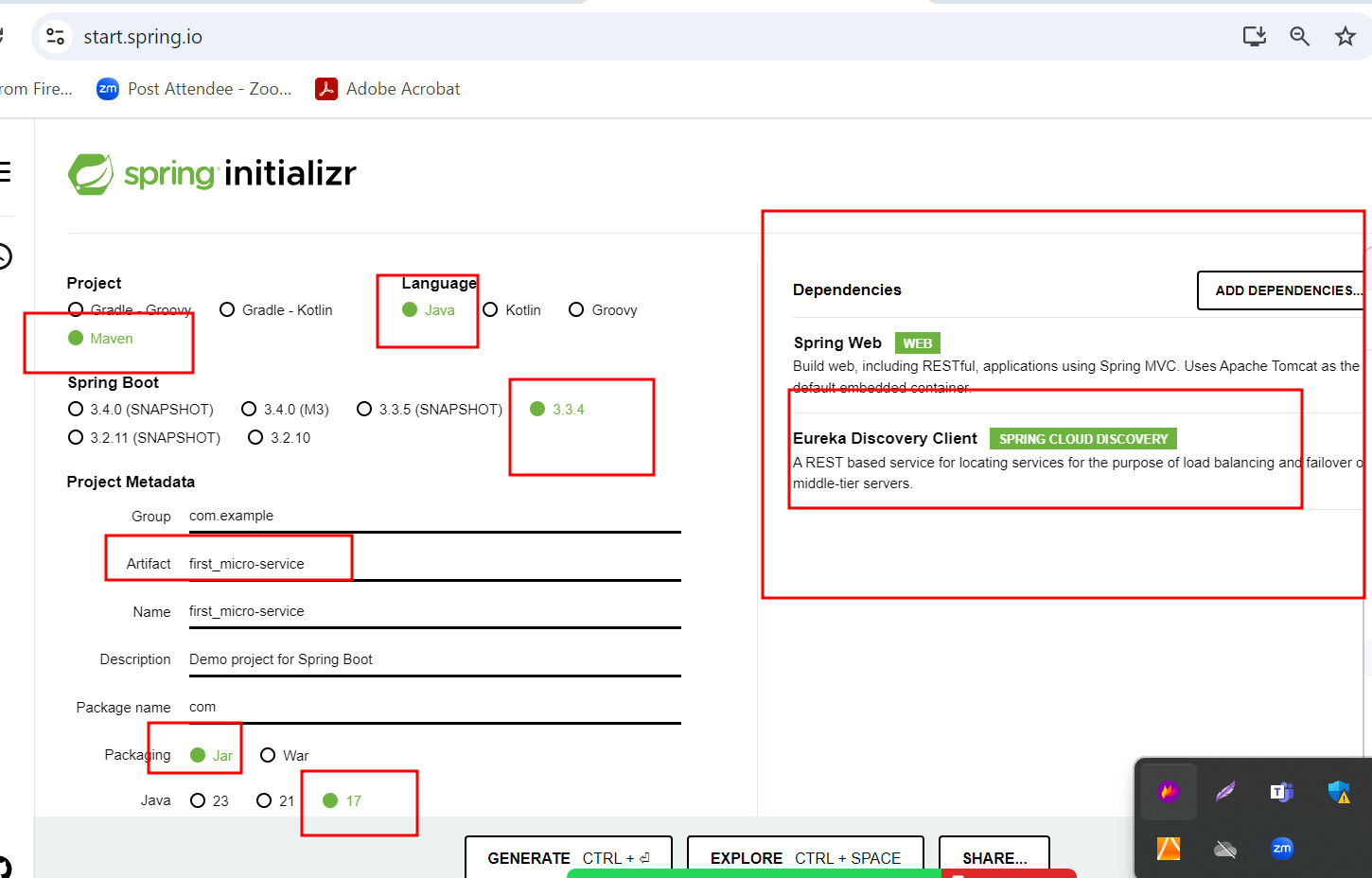
Now we can create more than one micro service.

2nd : first\_micro\_service : this project help us to create simple end point or rest api

Starter 🡪Web Starter

Starter 🡪Eureka Discovery client

Port number : 9191

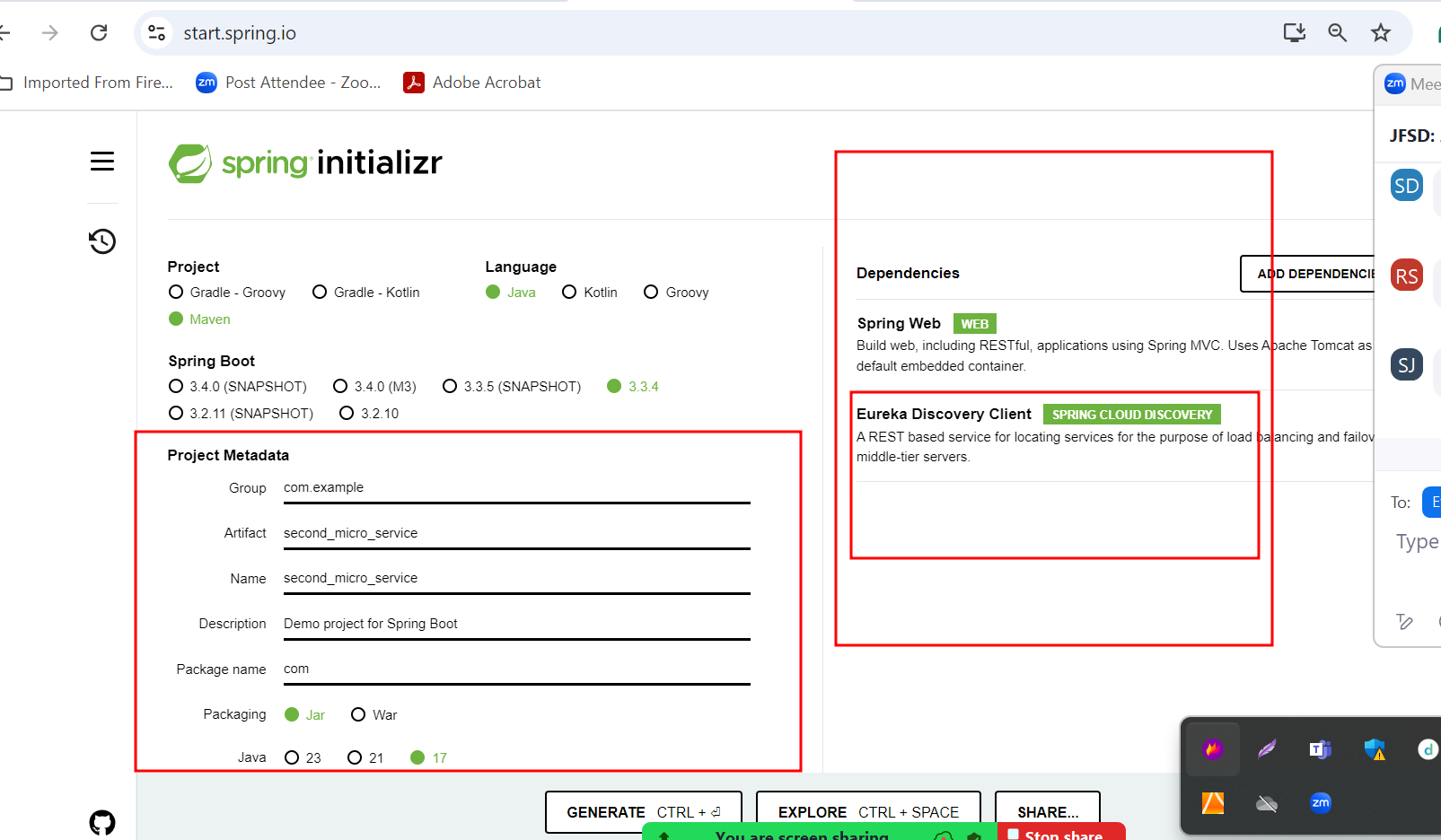


3rd : second\_micro\_service : this project help us to create simple end point or rest api

Starter 🡪Web Starter

Starter 🡪Eureka Discovery client

Port number : 9292



4th project

Account – micro service 9393

Starter -🡪Web Starter create rest api

Starter 🡪Eureka Discovery client : deploy on eureka server

Starter 🡪 JPA Starter : spring jpa data

Dependency 🡪 MySQL : allow to connect mysql db

Starter 🡪Dev tool : refresh the project whenever we do any changes

Starter 🡪 Lombok : it help use to avoid boiler plate code in entity class like setter/getter, toString(), empty constructor and parameter constructor

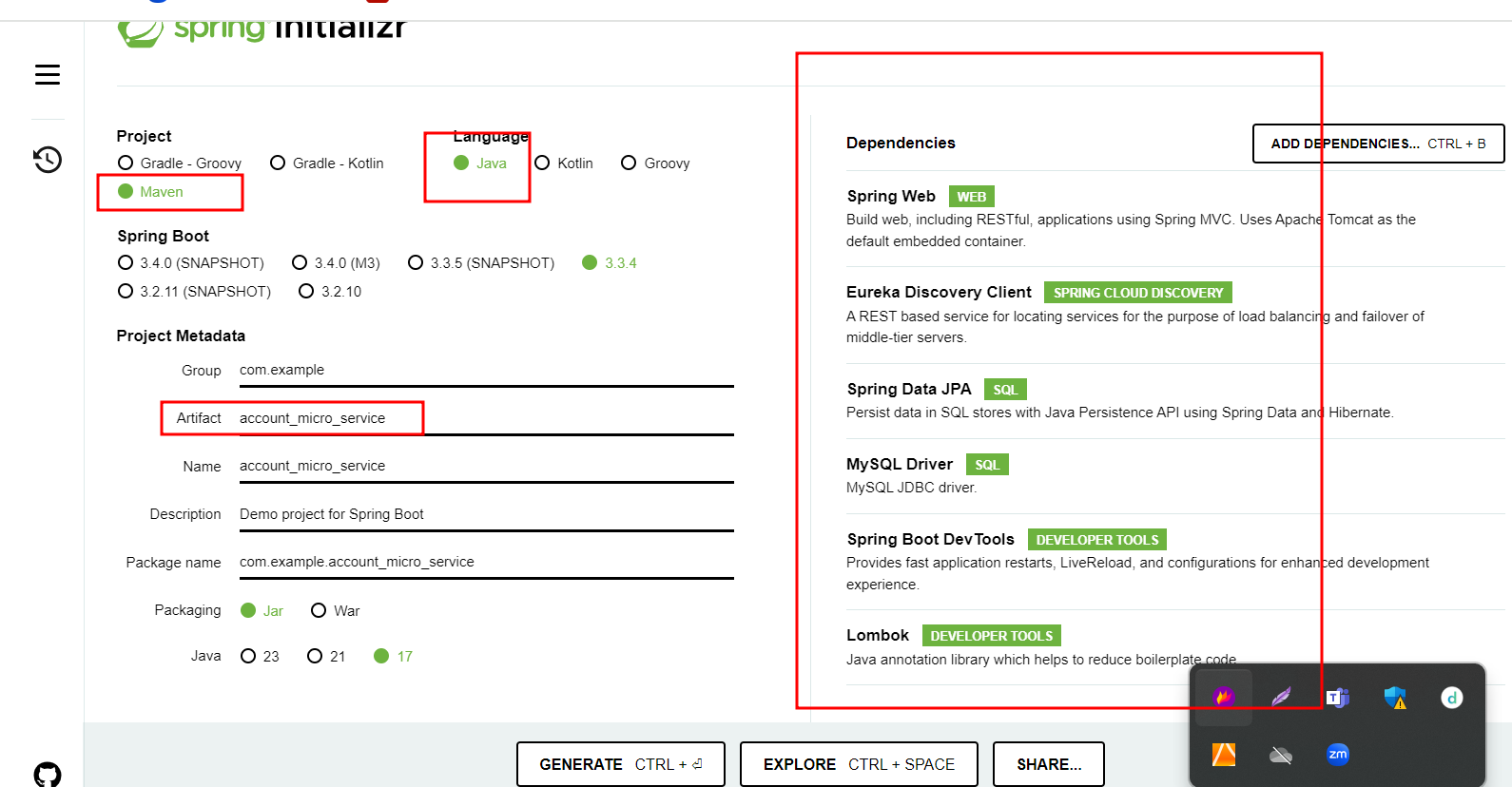
@Data it automatically provide use getter/setter, toString and empty constructor

@Entity

Rest api

accno(PK), name, amount, emailid(unique) etc

1. Create account with post method
2. Find account balance using accno and emailid (custom query)
3. Withdraw the amount
4. Deposit the amount



@RequestMapping : this annotation we can use on class level as well as method level.

Class level to map root path of specific controller.

Method level : we need to use value attribute to map sub path and method with RequestMethod.GET/POST/PUT/DELETE : generic for all http methods.

@XXXMapping annotation

Like

@GetMappiing

@PostMapping

@DeleteMapping

@PutMapping

@PatchMapping

These annotation we can use on method level without method attribute.

4th project

Account – micro service 9393

Starter -🡪Web Starter create rest api

Starter 🡪Eureka Discovery client : deploy on eureka server

Starter 🡪 JPA Starter : spring jpa data

Dependency 🡪 MySQL : allow to connect mysql db

Starter 🡪Dev tool : refresh the project whenever we do any changes

Starter 🡪 Lombok : it help use to avoid boiler plate code in entity class like setter/getter, toString(), empty constructor and parameter constructor

@Data it automatically provide use getter/setter, toString and empty constructor

@Entity

Rest api

5th project

Gpay – micro service 9494

Starter -🡪Web Starter create rest api

Starter 🡪Eureka Discovery client : deploy on eureka server

Starter 🡪 JPA Starter : spring jpa data

Dependency 🡪 H2 : allow to connect h2 db

Starter 🡪 Lombok : it help use to avoid boiler plate code in entity class like setter/getter, toString(), empty constructor and parameter constructor

@Data it automatically provide use getter/setter, toString and empty constructor

@Entity

Rest api

Post method ie postMapping

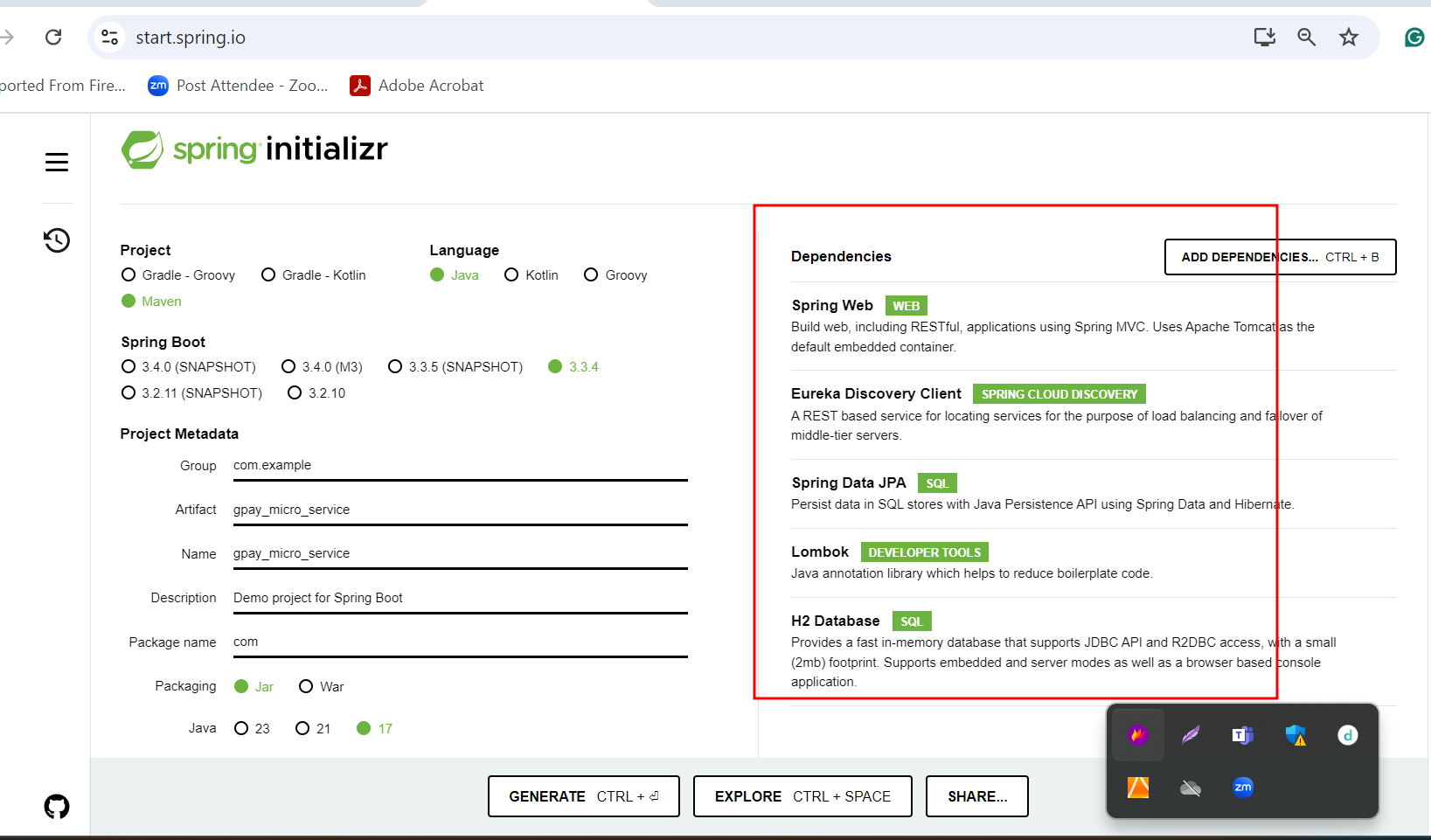
Gpid : auto increment

Emailid :

Get method ie getMapping

We pass gpaid to Gpay micro service. Using gpayid we get emailid. this gpay micro service interact with account micro service to get the balance.

9494 gpay micro service going to communicate with account micro service to get the amount balance.



Using Spring boot or micro service if we want to consume another rest api

1. RestTemplate API.