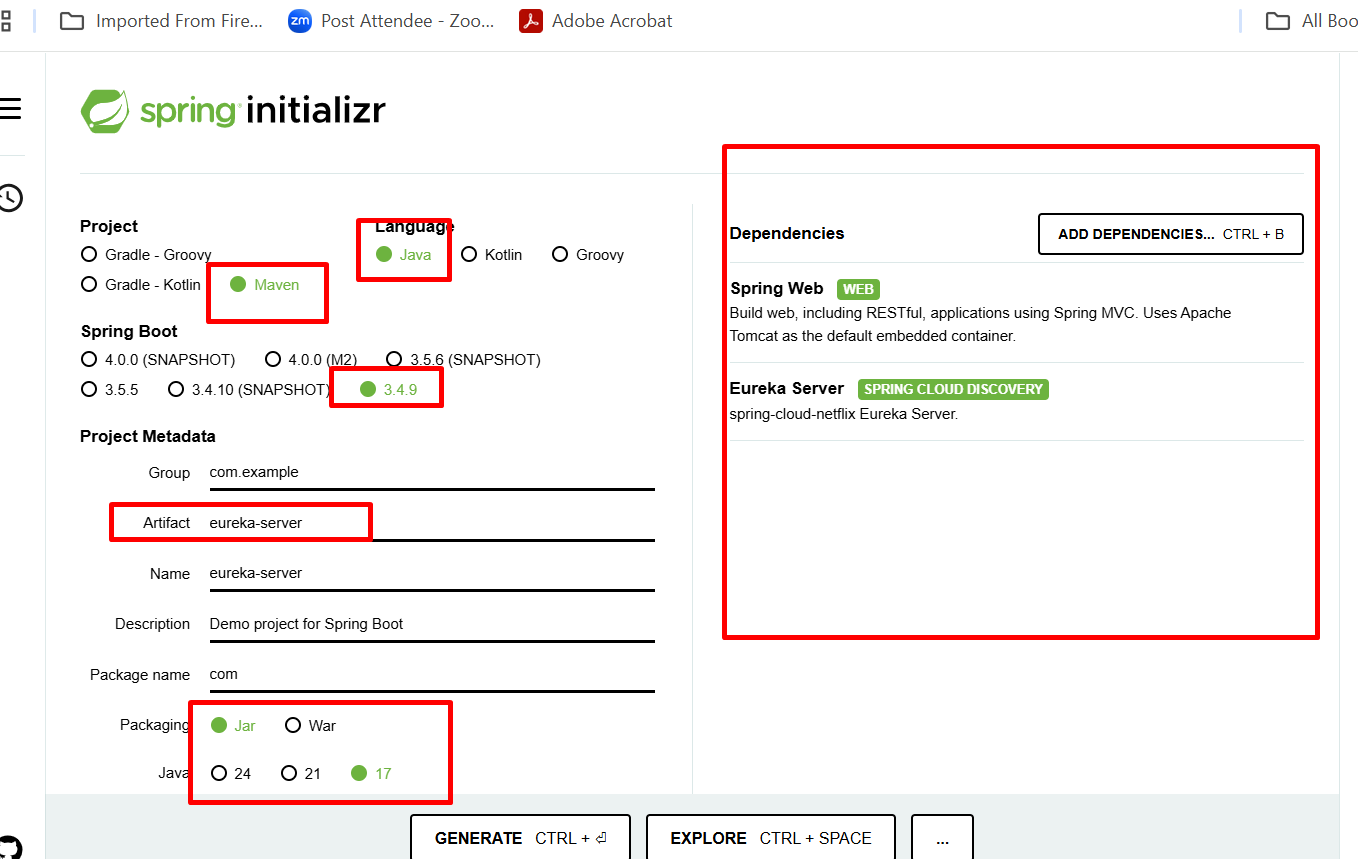
**Day 32:**

Creating micro service application using reactive approach with gate way. Call 2 micro service using WebClient.

1. Eureka Server --🡪 8761

Starter 🡪web starter

Starter 🡪 Eureka server



1. Product micro service 🡪 8181

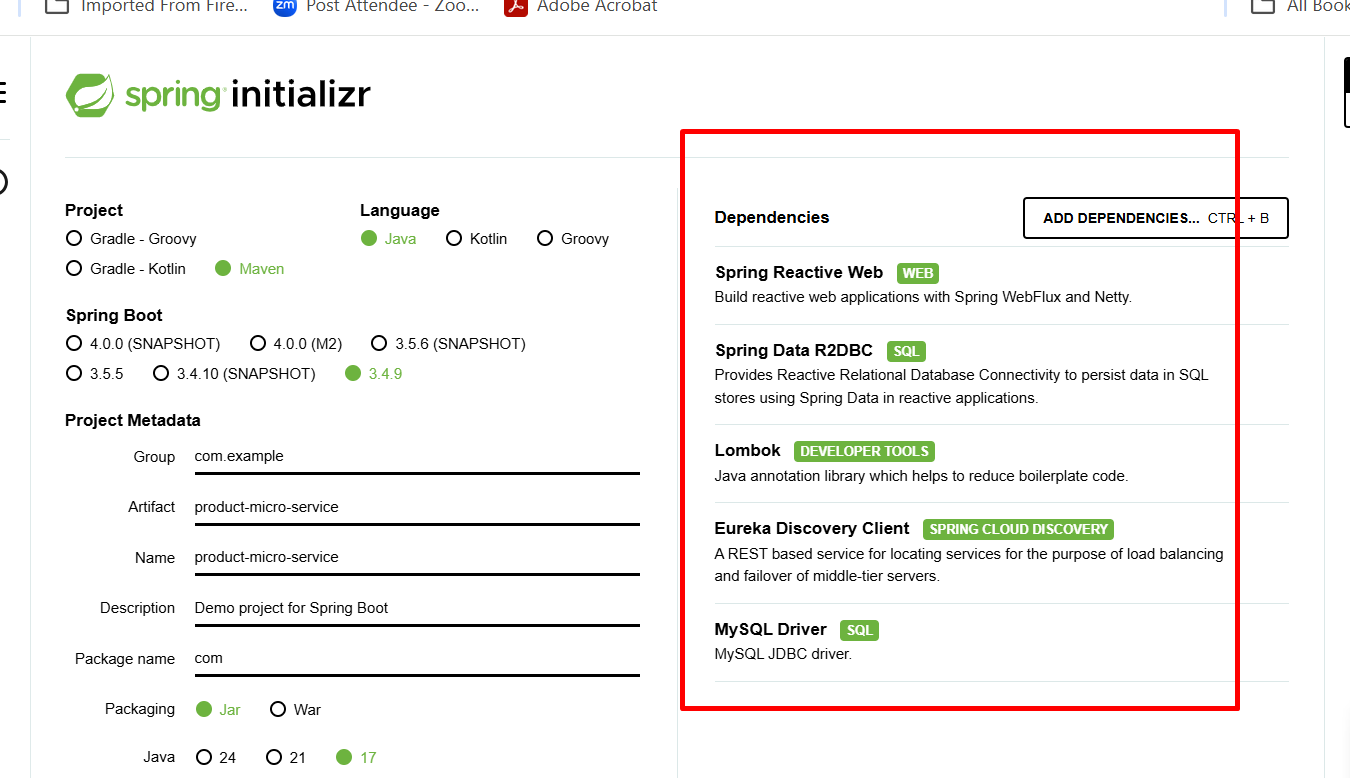
Starter 🡪 flux web starter

Starter 🡪eureka client

Starter 🡪 r2dbs

Dependency 🡪 mysql (we add manually)

Starter 🡪 Lombok



1. Order micro service 🡪 8282

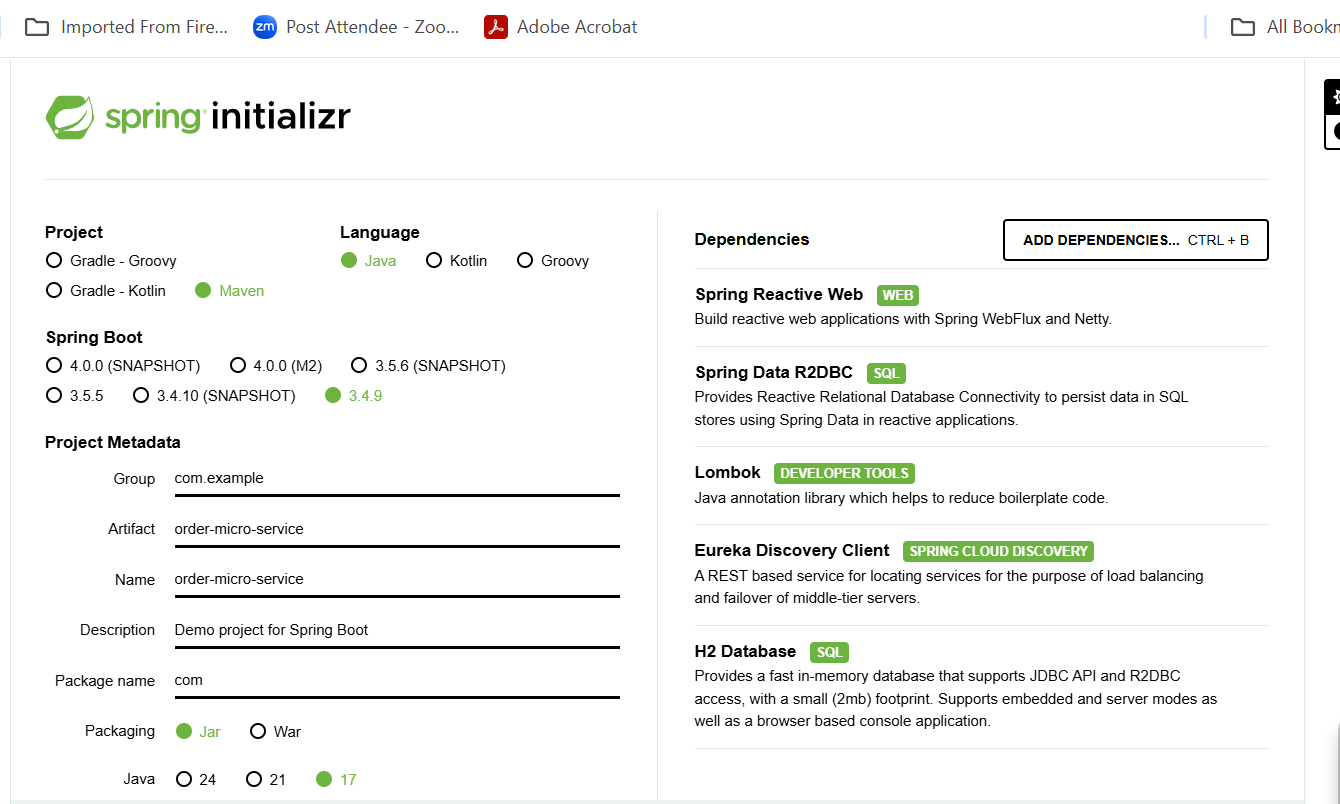
Starter 🡪 flux web starter

Starter 🡪eureka client

Starter 🡪 r2dbs

Dependency 🡪 h2 database

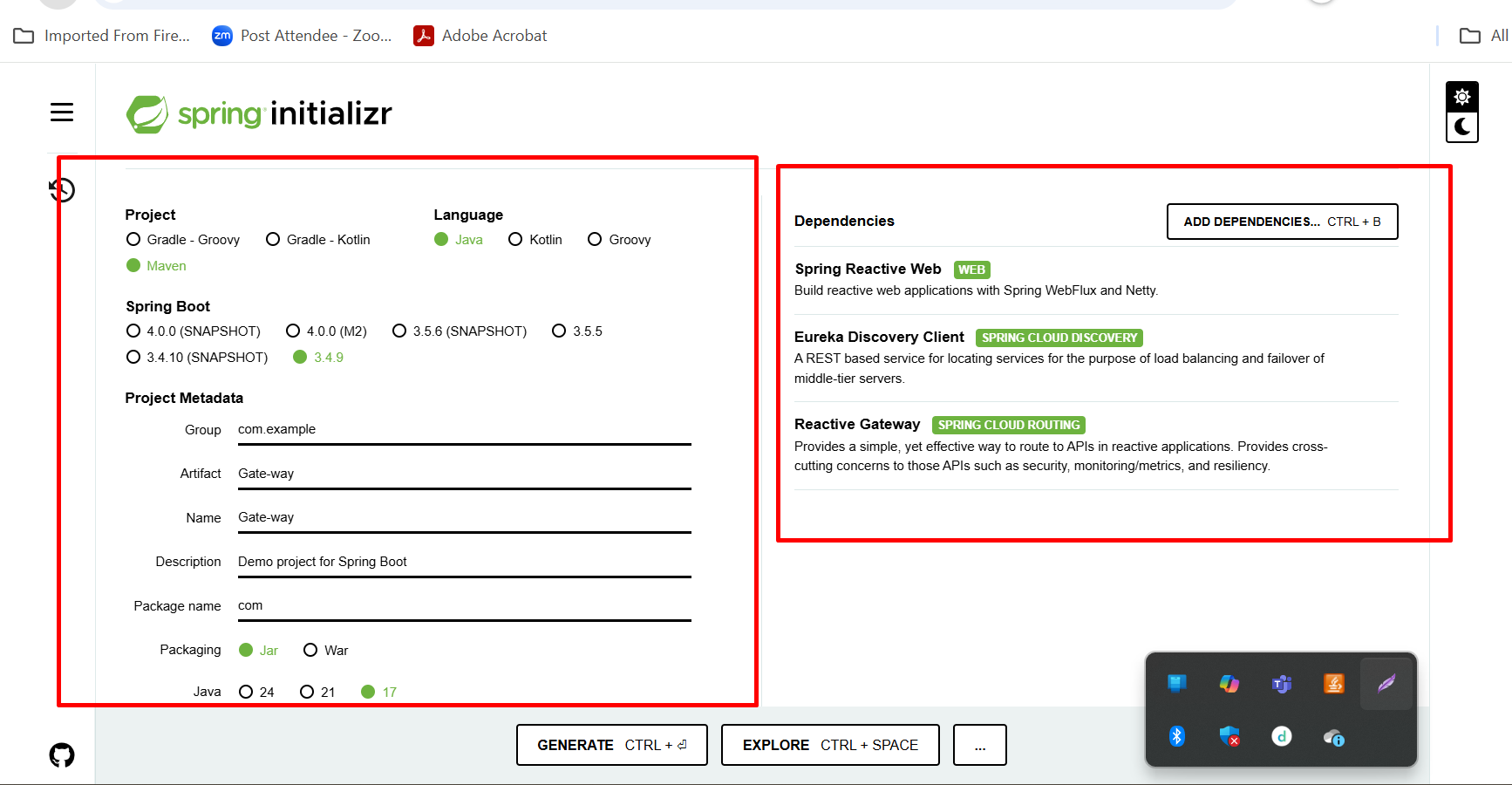
Starter 🡪 Lombok



1. Gateway 🡪8080 : main entry point for micro service to keep track.

Starter 🡪web starter

Starter 🡪gateway starter



The API Gateway takes all api call, from the client the application them to appropriate micro service base upon path or sub path.

**Spring Security :**

Spring boot provide starter which help achieve security features.

Security authentication and authorization

Authentication help us to use the resources ie access web page or access rest api.

Authorization mainly help us to access resources base upon the role.

If role is manager or admin. It can access all pages

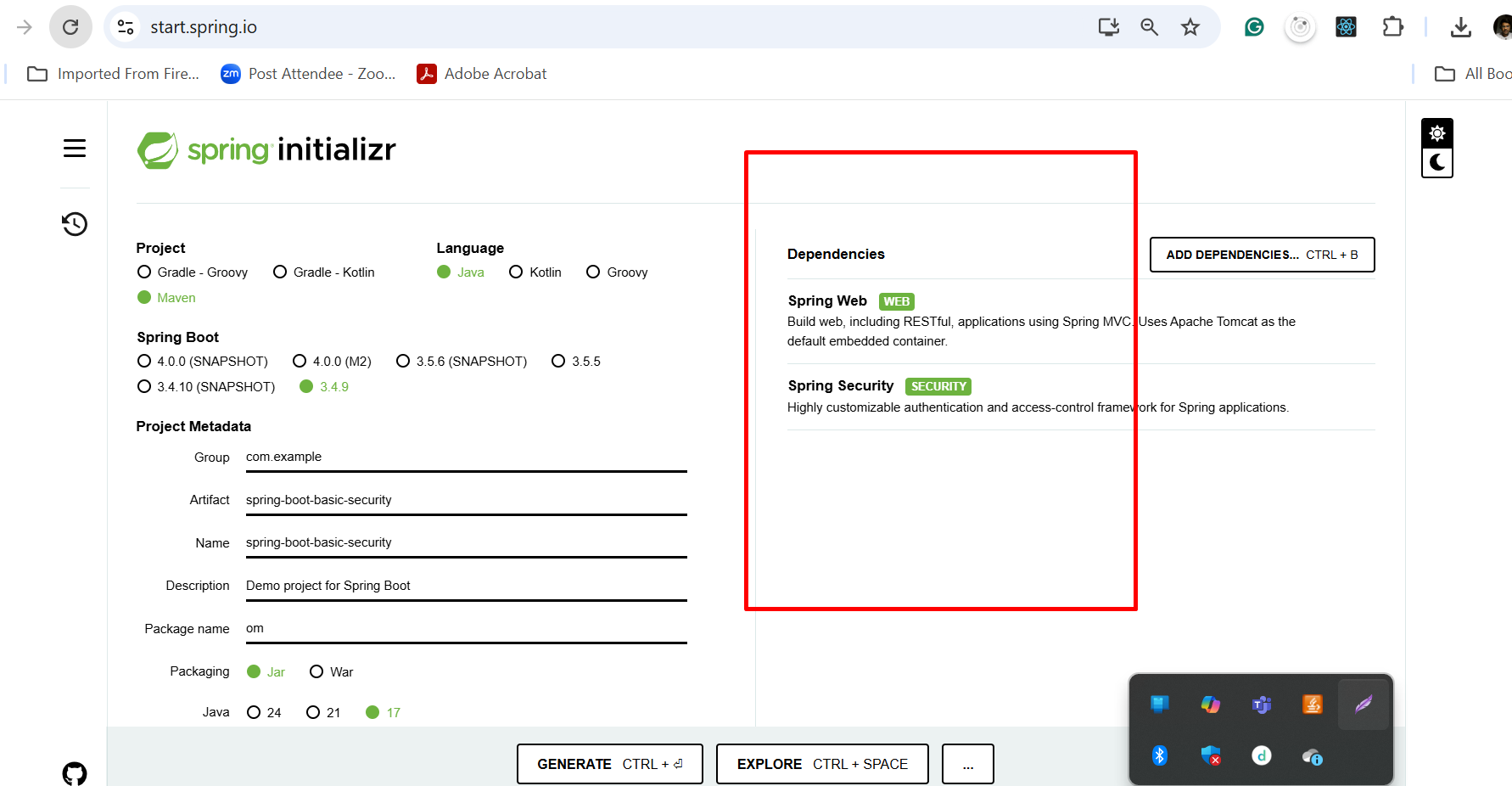
If role is user or customer it can access only few pages or end points.

Spring boot provide password encoded and decoded classes. Mainly store the password in hash format.

Spring boot support advanced level security features like JWT (JSON Web Token), oAuth 1.x oAuth2.x version.

**Simple Spring security example**

1. Spring boot starter
2. Spring boot security starter



Spring boot provide default user name is **user** password random generated. Which we can see on console.

If we want to provide custom user name and password.

1. Using application.properties
2. Using java class with @Configuration annotation.

Providing user name and password through java class ie configuration class.

Spring boot 2.x

Spring security classes are different

Spring boot 3.x

Spring security classes are different.