Phase 3

Day 1

06/03/2023

Creating maven project through command prompt

mvn archetype:generate

after type it will ask for the user to enter the information.

Please hit enter key.

After it will ask to version of the project

It will ask groupid : groupname

It will ask artifactiid : projectname

It will ask project version : here hit enter key

Package name : com

Then it will ask to create the project as Y.

After enter Y it will create simple maven project.

Maven GOAL

mvn validate : it check pom.xml file and verify all tags are correct or wrong.

mvn compile : this command is use to compile the project and it will create target folder. This folder contains all .class and build and files.

mvn clean : it will remove target folder.

mvn package : it will help to create jar or war file. (jar file for core java project and war file for web project).

Jar for core java project which contains .class file

War for web application which contains servlet, jsp, xml, html, css and javascript etc. which we need to deploy on server.

Testing : Testing is use to find the defect or error or bugs in the application.

MVC

To run out application layer without main method is known as testing.

Service Layer :

Testing : mainly divided into 2 types.

Block box testing

Input Process output

White box testing

Input Process Output

Unit Testing : Unit testing is a type of white box testing which help to test function functionality working or not. Unit is smallest work which work independently. Ie inside method or function or modules.

jUnit is open source light weighted framework which help to do unit testing using Java technologies.

Test NG (next generation): it use for unit testing as well as integration testing.

nUnit : for .net

Jasmine : for HTML,CSS,JavaScript as well as Angular

React Js : JEST

Node JS : Mocha with Chai as well as Jasmine.

jUnit : 3.x (without annotation)

4.x with annotation

5.x with annotation with combination of more than one third party framework with Java8 features.

Test case : In Java test case is a type of junit test case class which contains more than one test function which help to test function functionality working or not.

We need to create normal java class with method with annotation as @Test.

Test suite :

Junit provided lot of pre defined assertXXX type of method which help to check expected and actual output.

jUnit hook :

4 hook we can use

@BeforeClass

@AfterClass

@Before

@After

Test Suite : It is a type of test class which is responsible to execute more than one test case class.

Mvn test : command is use to test through maven through command prompt.

Phase 3

Day 2

06/04/2023

Spring Framework

Framework : Framework contains set of API (Application Programming Interface). Like it contains set of classes and interfaces which internally connected to each other to do specific task.

Design pattern : Best practise or solution for repeating problem.

The implementation of all design pattern is taken care by framework. If we develop any application using any framework indirectly we are following standard. Framework do 70 to 80% task of our project or application. But framework is not final product it a protocol or template.

.net framework. Which support more than 45 language like C#, Asp.net Vb, J# etc.

Java technologies

Java open source framework.

Struts : Struts is an open source web framework provided by apache. Struts internally follow MVC architecture. It provided lot of API to improve controller layer, view layer and model layer. It internally follow Front Controller design pattern. Struts is known as Controller centric framework.

JSF : Java Server Faces : JSF is an open source web framework provided by oracle. JSF is replacement of JSP. JSF internally follow MVC design pattern. Provided lot of api to improve view layer, controller layer and model layer. It internally follow front controller design pattern. JSF is known as View Centric framework.

JSF Vs Angular or React JS

Hibernate : Hibernate is an ORM framework (Object Relation Mapping). Hibernate is replacement of JDBC. Using Hibernate or JPA(Java Persistence API) we can improve DAO Layer.

Spring framework : Spring is an open source layer or module architecture framework. Spring provided lot of modules or layer to improve all types of application. Spring is light weighted framework.

Spring modules

Spring core

Spring context

Spring MVC : spring mvc internally follow mvc design pattern. Provided lot of api to improve controller, mode and view layer. Spring MVC is known as Model Centric framework.

JEE

Servlet Controller

JSP View

EJB Model

EJB Vs Spring MVC

Spring REST

Spring DAO

Spring ORM with Hibernate or JPA

Spring security

Spring boot

Spring cloud

Spring micro service

Etc

JEE : MVC project with View as HTML /JSP and Controller is servlet.

JSP and Servlet object creation taken care by container.

Model layer object we are creating explicitly. Means Login bean, LoginService and LoginDao.

To improve model layer we use EJB.

So if we give model layer to EJB developer then can maintain the model layer properly.

If we want to run ejb program we need to ejb container. EJB container is a part of application server.

EJB is very weighted components.

Spring Framework

IOC : Inversion of control : IOC is a concept or programming design pattern. According to IOC in place of creating or maintaining any resource like object creation, security, database connectivity explicitly allow to create or maintain by container. If container create then it will maintain properly rather than once. You need to pull from container whenever you required use it and leave it.

DI : Dependency Injection

DI is a implementation of IOC. We can achieve DI using

Constructor base

Setter base

To do di using constructor base or setter we need to configure using XML or annotation.

SpringDIUsingXML

Web container will create the object of that class if class extends type of servlet or file must be jsp.

EJB container will create or maintain the object if class is type ejb.

Spring framework create the object of normal java class ie POJO. (Plain Old Java Object).

The class not to extends or implements any special class normal class ie JavaBean class.

If container create the object using xml by default it consider as singleton.

If you want each time new memory whenever we pull using id then we need to scope as

Prototype.

To achieve setter base di in pojo class we need to write mandatory setter method.

Constructor base DI support only fully dependency

Ie default constructor or parameter constructor with number of parameter and type of parameter must be match.

Setter base constructor support fully dependencies as well as partial dependencies.

Auto wiring

By default spring framework (spring container) do the di for primitive property with default value. If class contains complex property then we need to do explicitly to do di using property ref or constructor-args ref. Rather than doing explicitly di for complex property we can use auto wired features. Auto wired is spring framework features it automatically do the di for complex property implicitly rather than explicitly.

To do autowired using xml we need to set autowired with different values.

byType : if we set byType spring container scan whole xml file to check type of bean definition in xml file.

If we use byType in xml file we need to provide only one bean definition of that type.

If we write more than one bean definition of that type then we need to use byName.

In byName reference name in bean class and id name must be match.

Spring DI Using annotation

@Component : this annotation we need to write on pojo or java bean class.

@Autowired : this annotation we need to use on complex property.

By default @Component annotation is not enable so we need to enable using

Xml file

Using java class.

@Value annotation : this annotation to set default value for property

Spring Framework with JDBC (improve dao, service and resource layer)

Dependencies added

Core

Context

Jdbc

Mysql connector

Spring framework provided pre defined class

Ie DriverManagerDataSource part of spring jdbc dependencies

This class is responsible to provide database connection in secure manner with

Singletone features.

@Component :Generic annotation which we use on normal class

@Repository : This annotation we use on dao layer. This is dao layer specific annotation.

@Service : this annotation we use on service layer.

Spring provided JdbcTemplate :

JdbcTemplate is a pre defined API provided by Spring Framework which internally

wrap core JDBC Code and provided set of methods which help to improve dao layer.

Git pull

You are in master/main

git branch springdao

git checkout springdao

git checkout master/main

git merge springdao (everything merge in master).

git pull conflict

git branch -D springdao

Dev1

Dev2 shared repository

Dev3

Limitation of JDBC or JdbcTemplate

1. Using JDBC we can’t store and we can’t retrieve java object. We need to convert

Query to java object and vice-versa.

1. JDBC use SQL language . SQL is database dependent language. When we use from one database to another database like mysql to oracle query get changed.
2. Jdbc throw checked exception. Exception hierarchy in Jdbc is database dependents.
3. JDBC doesn’t support relationship. Like is a (inheritance ) has a (aggregation and composition).

ORM : Object Relation Mapping

According to ORM in Programming side object (bean class) and database table (relation) and we are doing mapping.

Object Table

class Product { Product

pid,pname,price PId,PName,Price

}

Mapping

Product ---🡪PRODUCT

Pid-🡪PID (PK)

PName🡪PNAME

Price 🡪PRICE

This mapping we can do using XML or annotation

Phase 3

Day 4

06/11/2023

Hibernate : Hibernate is a third party framework which provided ORM features.

JPA : Java Persistence API. JPA is technologies which support ORM features.

JPA is a specification as well as it provide implementation.

Hibernate is a third party framework base upon JPA. Hibernate provide implementation of JPA.

JPA is a type of EJB.

JPA alone with Core Java Project

Hibernate alone with core Java project.

Servlet /JSP with Hibernate or JPA for Web Application

Spring Framework provided Spring ORM (Object Relation Mapping).

Spring ORM provided lot of classes and interfaces which help to interact with JPA or hibernate.

Spring Boot : with Spring boot we can use JPA but not Hibernate.

JPA : Java Persistence API

JPA use persistene.xml file to store Database details like drivername, url,username and password.

@Entity on java bean class

@Id the column which contains PK.

@Table : if class name and table name can different then using @Table annotation we can

Map.

@Column : if variable name and column name different then using @Column

annotation we can map

By default if we do any operation using JDBC it is auto commit. Maily DML Operation.

By default if we do DML operation using ORM tool like JPA or Hibernate it not auto commit

So we need to transaction.

OMR provided their own query language like

SQL : SQL is database dependent. Using SQL we can retrieve records in string format.

Select \* from employee (where employee is table name and sql is not case insensitive).

Select \* from employee where salary > 14000;

Select name from employee

Select id from employee

Select id,name from employee;

JPQL (in JPA) or HQL (In Hibernate) : JPQL is database independent.

Select emp from Employee emp (Here employee is entity class name and it is case sensitive and

Emp is reference name)

Select emp from Employee emp where emp.salary > 14000;

Here emp is object and salary is variable name in Employee class.

Select emp.name from Employee emp;

Select emp.id from Employee emp;

Select emp.id,emp.name from Employee emp;

ORM Relationship

JPA support relationship

One to One Person has one passport

One to many One trainer handle more than one student.

Many to one Many employees working in one project

Many to Many Many students known more than one technologies.

One means PK and many means FK

One to many relationship

create table trainer(tid int primary key,

tname varchar(25),

tech varchar(25));

create table student(sid int primary key,

sname varchar(25),

age int,

tsid int,

foreign key(tsid) references trainer(tid));

Phase 3

Day 5

06/17/2023

Spring MVC (Model View Controller)

Normal Servlet Vs Spring MVC Servlet

class Login extends / implements Servlet/GenericServlet/HttpServlet {

we need to override life cycle method as well as doGet or doPost etc

We need to create PrintWriter class object

}

We need to provide servlet configuration in web.xml file or using annotation.

@Controller

class Login {

@RequestMapping(value=”hello”,method=RequestMethod.GET)

public ModelAndView openPage() {

// do our work

ModelAndView mav = new ModelAndView();

mav.setViewName(“display.jsp”);

return mav;

}

public String doTask () {

}

}

Spring MVC internally follow another one design pattern ie FrontController

ProducController , CategoryController, LoginController, AccountController

FrontController is a type of servlet which control all controller flow.

Spring framework internally provided front controller pre defined class ie DispatcherServlet.

Which we need to configure using web.xml file or normal java classes with annotation.

View -🡪 index.jsp --🡪web.xml file (DispatcherServlet is front controller is configured).

Here DispatcherServlet search spring configuration file start with name as servletname-servlet.xml

Abc-servlet.xml

dispatcher-servlet.xml

Then Spring container scan com package which contains

@Controller, @Service, @Repository as well as @Component annotation classes

And it will do the DI for those classes.

Spring MVC with Database (@Service, @Repository and Autowired for DataSource etc).

In dispatcher-servlet.xml we need to provide DriverManagerDataSource class details.

Spring MVC with Database using JdbcTemplate (@Service, @Repository and Autowired for JdbcTemplate etc).

In dispatcher-servlet.xml we need to provide DriverManagerDataSource and JdbcTemplate configuration.

Spring core

Spring context

Spring jdbc with JdbcTemplate

Spring MVC

If we develop any application using spring framework with help of one more

Than one module we need to do more configuration using xml file or using annotation.

Spring boot : Spring boot is like a bootstrap for all spring modules. Spring boot not to develop any specific application

If we develop spring any module depends upon our requirement with help of spring

Boot we need to write very less configuration.

Spring boot = All spring modules – no xml file + few annotation + inbuild or embedded tomcat or jetty server.

CSS Vs Boostrap

Using spring boot we can develop ready made application using RAD (Rapid application Development).

Spring boot with build tool

1. Maven : if we use maven then we need to use pom.xml file (it is a type of xml file)
2. Gradle if we use Gradle build tool then no xml file.

Spring boot modules

1. Spring boot starter :

Maintaining the dependencies. Jar

The main responsibility for starter is to combine all common or related jar into jar file.

Web Starter

Jdbc strarter

Jpa stater but no hibernate starter

Spring testing starter

Spring security

Etc

Spring boot itself is core java or standalone project spring boot is not web project is normal

Core java project but it will help to develop any type of project.

In spring boot we will use main class in development mode.

1. Spring boot auto configuration

Before spring boot we were doing all configuration using xml or annotation.

But spring boot provide auto configuration features. Base upon what type of starter we added in project automatically they provide those resource for us.

Spring boot provided one pre defined annotation ie

@SpringBootApplication : it is type of class base annotation which we need to write on class which contains main method.

@SpringBootApplication = @Configuration + @ComponentScan + @AutoConfiguration

Spring boot we are not doing any configuration using xml file.

But if we want any resource details like dataset or security then we need to write in application.properties. in this file we write information using key-value pairs.

Spring boot internally using in build tomcat server default port number is 8080.

Spring boot doesn’t support or didn’t provide JSP starter.

If we need to view as JSP then we need to add jsp dependencies.

Or

Spring boot with View as Thymeleaf engine

HTML Vs JSP Vs Thymeleaf

Html is static

Jsp covert to servlet

Thymeleaf it dynamic java engine.

@Controller annotation then view must be jsp or thymeleaf

@RestController annotation then view can be any technologies like angular, react, java or python etc.

@RestController = @Controller + @ResponseBody;

@SpringBootApplication : This annotation by default scan all the classes present in same package or sub package of current package.

If we need to scan other packages then we need to use attribute as scanbasepacke part of @SpringBootApplication.

Phase 3

Day 6

06/17/2023

Get Method : Retrieve Resources select query or select query with condition

1. Retrieve string data in plain text, xml, html etc.
2. Get resource details like Employee, Customer, Product etc. JSON/XML
3. Get all resources details like List of Employee, Customer or Product etc. JSON/XML
4. Get with Query param
   1. Single query param :URL?key=value
   2. Multiple query param : URL?key=value&key=value

By default Plain html with form with get method internally use query param technique.

1. Get with Path param
   1. Single path param : URL/value1
   2. Multiple path param : URL/value1/value2/value3

Post Method : create Resource : insert query

1. Store the data or create the resource.

Put method : update Resource update query

1. Update record

Delete method : delete resource delete query

Spring boot with database connectivity

1. Core JDBC
2. JdbcTemplate
3. ORM (JPA)
4. Spring Data.

Hibernate internally use JPA annotation.

JPA Hibernate

EntityManagerFactory SessionFactory

EntityManager Session

EntityTransaction Transaction

Persistice save

Remove delete

Merge update

Find get

Spring boot with JPA (ORM)

Web starter

Jpa starter

Mysql connector dependencies

In JPA project we configuration database details in persistence.xml file

In Spring MVC with JPA we need to configurate in dispatcher-servlet.xml file.

In Spring boot we need to provide database deails in application.properites

Spring Data :

Spring ORM doesn’t provide any orm tool. They allow us to integrate with existing orm tool like

JPA or hibernate or iBaties or Jdo etc.

Spring Data is spring module which is base upon JPA. Spring JPA Data.

According to Spring JPA Data rather than writing code in DAO layer from beginning we need to create one interface and that interface must be extends JpaRepository. JpaRepository is base upon java 8 features.

@Repository

public interface EmployeeRepository extends JpaRepository<EntityClass,PrimaryKeyDataType> {

JpaRepository provided lot of pre defined method with internal logic to do all Standard operation on that entity.

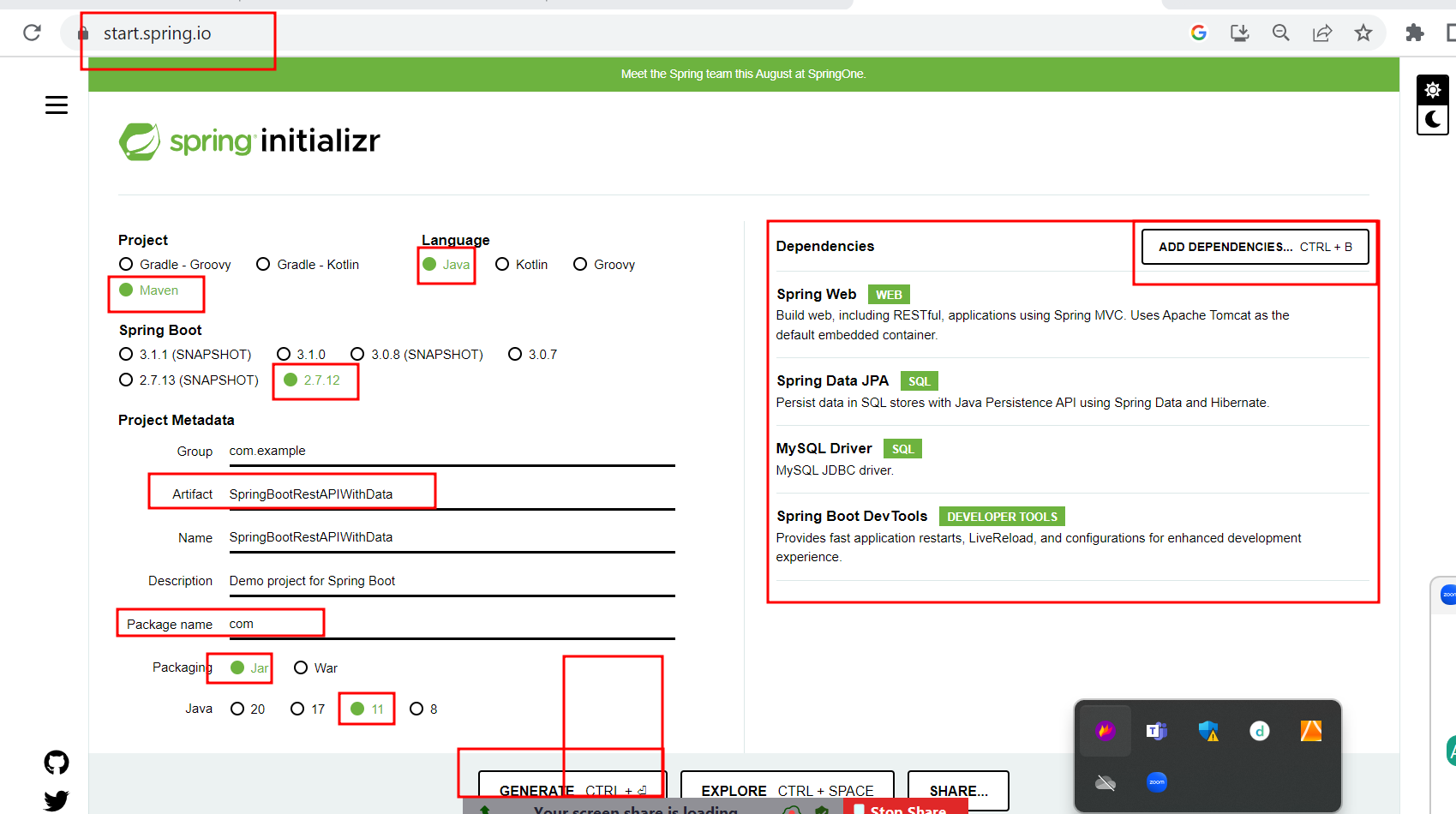
Save, delete, update, retrieve with lot options.

}

JPA repository doing abstraction for DAO layer.

If we want to write some custom query depending upon our requirement then we need to use JPQL or native sql also with @Query annotation.

Spring Rest API with Spring Data with creating project with spring initlizer.



Phase 3

Day 7

06/24/2023

Micro service : small services running independently.

Monolithic Vs Micro Service

Monolithic:

Product : Product Entity , Product Controller, Product Service , Product DAO or repository , database(table)

Category

Account

Login

Manager

Customer

Orders

We need to create single jar or war etc.

Micro Service :

Product Related task using Java with MySQL Database we can deploy this project independently with

Product related jar or war file.

Category related task using Java with oracle database we can deploy this project independently

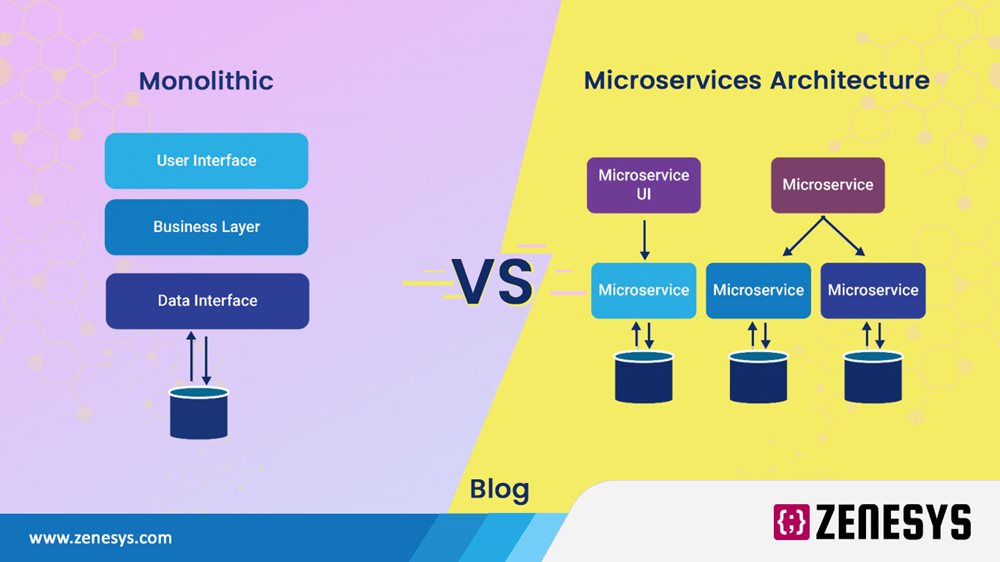
Category related jar or war file.

Login we can develop using node js with mongo db or mysql database and deploy this project

Customer we can deploy using python and deploy with any other database postgres SQL.

The interaction between more than one module using same language or different language

With help of rest api.





Micro service is concept.

To achieve micro service in java with help of Spring boot

Spring provided two modules

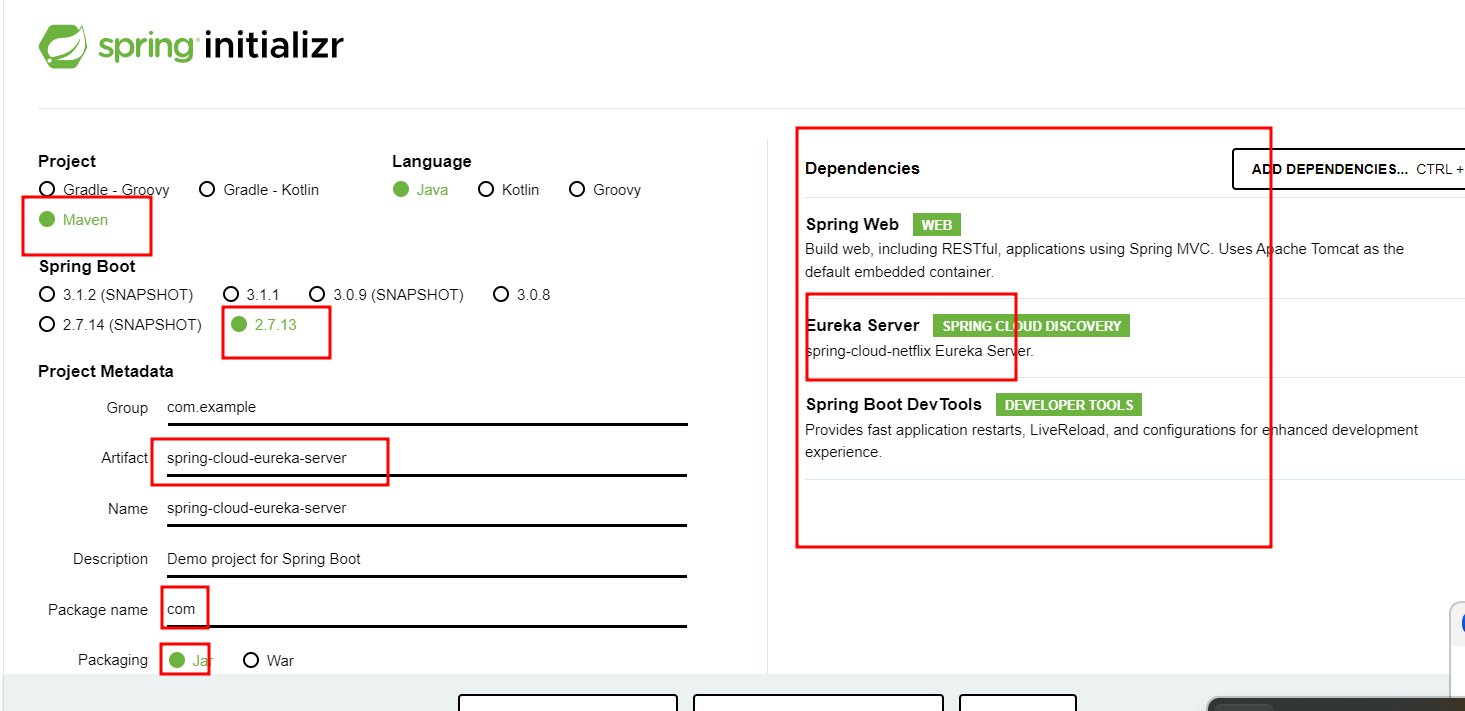
Spring cloud

Spring micro service

Once we created multiple micro services we need to deploy these all micro services project. Spring framework provided another server is Eureka. It is an open source server provided by spring cloud which help to deploy more then one micro service running using java technologies.

If we need to implements micro service concept using spring boot

First we need to create one project with web starter and eureka server starter. This project help use to deploy more than one micro service project.



By default every spring boot project internally run on tomcat server with default port number as 8080.

In Eureka Server project we added web starter as well as eureka starter so eureka server run on default port number 8080.

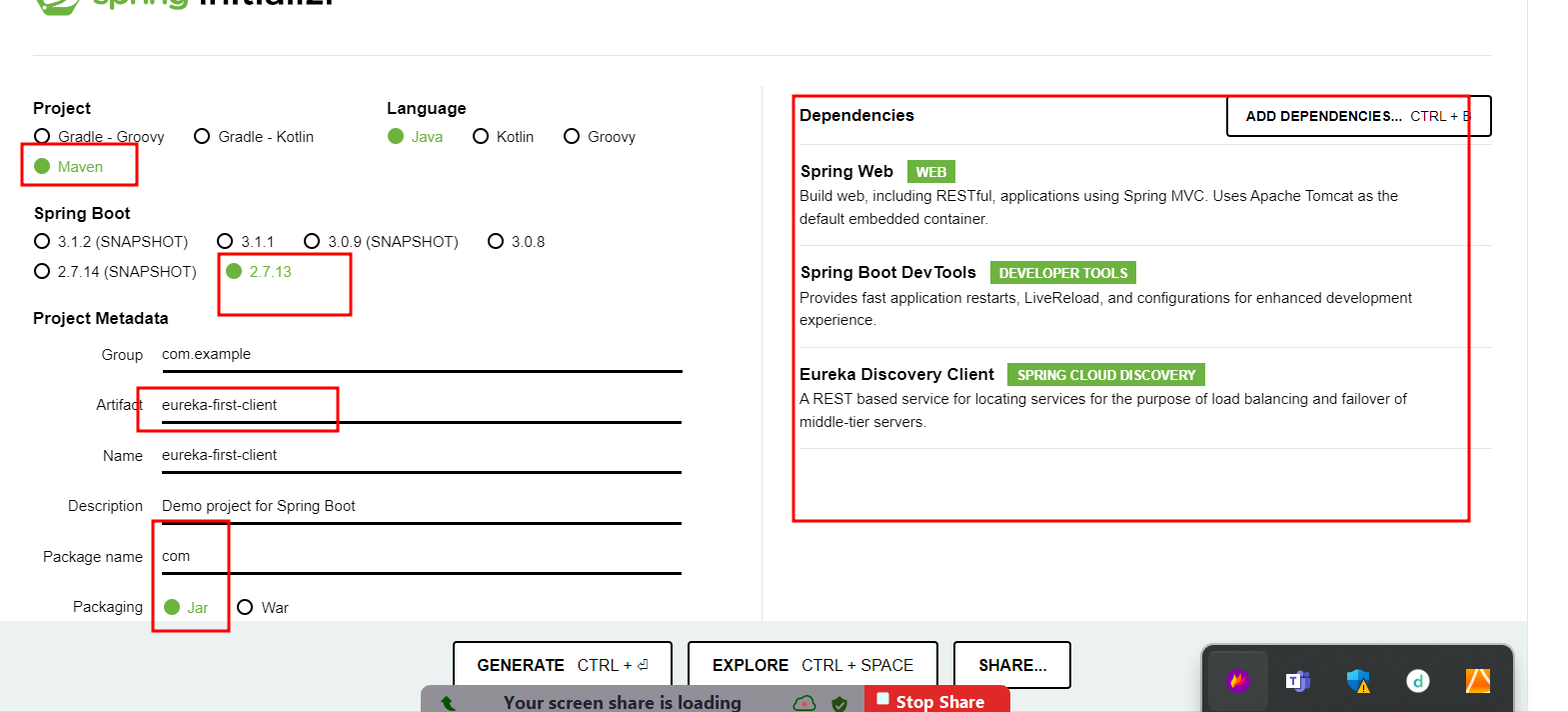
Then we create more than one micro service project. All project by default run on default port number 8080. So we need to change the all micro service project port number as 8181, 8282, 8383 etc.

If we create spring boot project with eureka client starter. By default eureka client search eureka server on default port number is 8761( 8761 is default port number for eureka server).

By default eureka sever project internally consider as client application. This application it behave like server as well as client.

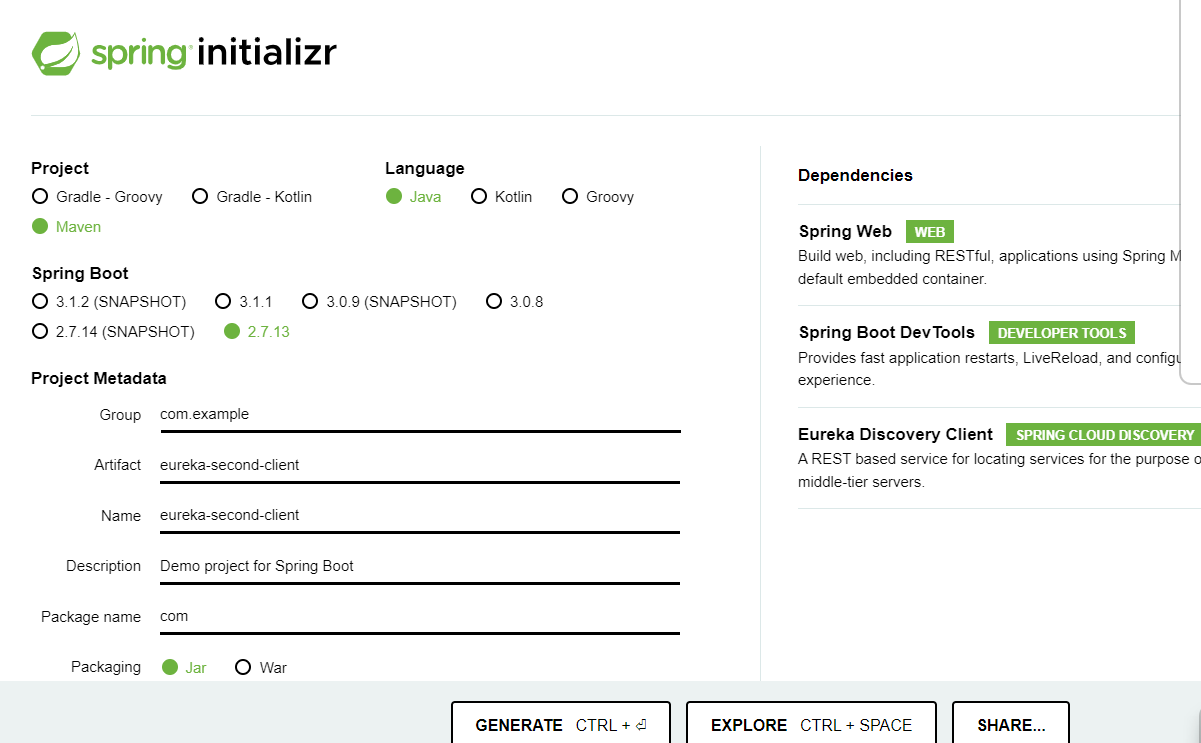
Create first-client as (Eureka Client Application)

Starter : Web, Eureka Client, Devtool etc



Create second-client as (Eureka Client Application)

Starter : Web, Eureka Client, Devtool etc



Third micro service project with web, mysql connector, jpa starter, dev tool and eureka client.

Product Micro service 8282

Customer micro service 8383

Customer micro service behave like a consumer for Product micro service.

Customer, Product, Manager or Login.