Phase 3

Day 1

18-03-2022

Spring Framework

Spring boot

Junit 5

Web Service (RestFull Web Service)

MVC : Model View controller

View - HTML/JSP : presentation

Controller -- Servlet : middleware between view and model

Model --

JavaBean or Entity class : container class or link with table

Service class : pure business logic

Dao class : pure jdbc or orm (hibernate code)

Resource class : database details.

Xml --- database connection

Limitation of MVC project created using servlet, jsp and normal model layer.

Web container create the object of servlet and jsp but not model layer. Model layer object we are creating. When we create the object we have to maintain the life of the object.

Web container part of tomcat server will tell. I will create the object of those class which must be type of servlet or jsp.

To improve the model layer we can use EJB. EJB is a use to improve the model layer.

EJB : Enterprise Java Bean.

What EJB

EJB Vs Spring

If we provide Model layer to EJB. EJB maintain creation of Entity or JavaBean class object, service layer, dao layer and resource layer.

To run the EJB application we require EJB container. EJB container is a part of Application server ie web logic or jboss or glashfish etc.

EJB is heavy component or complex to develop the application.

Type of EJB

Session Bean

Entity Bean or JPA (Java Persistence API ORM)

Message Driven Bean

Spring Framework is replacement of EJB.

Spring Vs EJB

Difference Hibernate and JPA :

Framework : Framework provide set api which internally connected to each other to perform a specific task. Framework follow standard. Framework is a implementation of design pattern.

When we develop any application using framework. The framework internally take care 70% to 80% task so we have to write 20% to 30% core to make the final product. Because framework is not a final product it a protocol or template.

Design pattern : Why, When and Where.

Best practise or solution of repeating problem.

Java technologies

.net framework

Struts : Struts is a open source web framework provide by apache. Which internally follow MVC architecture. It provided lot of classes to improve view layer, controller and model layer. Which internally follow front controller design pattern. ActionServlet

Struts is known as control centric framework.

JSF Java Server Faces : JSF is open source web framework. SF is replacement of JSP. JSF is a part of oracle company. It follow MVC architecture. It provided lot of classes to improve view layer, controller and model layer. Which internally follow front controller design pattern. FacesServlet

JSF is known as View centric framework.

Spring : Spring is a open source light weighted layer architecture or onion framework.

Spring framework provided lot of module which help to improve every layer application.

Spring core

Spring context

Spring Web

Spring MVC : provided lot of api to do MVC application provide classes to improve view layer, controller layer and model. Spring MVC is known as Model Centric framework.

Spring Rest

Spring DAO : Data access object using JDBC

Spring ORM : Spring with Hibernate or JPA

Spring AOP : Aspect oriented programming like Filter.

Spring Security

Spring testing

Spring cloud

Spring boot

Spring micro service

Spring integration with other framework

Etc

Hibernate

Spring core

IOC : Inversion of control

IOC is concept or design pattern. Rather than creating the object or resource explicitly allow to create and maintain by container. If container create it will maintain properly.

In place of creating any class object explicitly. Allow to create by container.

Web Container will create the object of that class if class is type of servlet or jsp or Struts class or JSF.

Spring container will create the object of class not mandatory class must be type of special class. class can be POJO (Plain old Java object). it is normal class not to extend or implement any pre-defined class. Spring container is light weighted it is part of jar files.

Web Container and EJB container they are heavy because they are part of server.

DI : Dependency Injection

DI is a implementation of IOC. To implements IOC concept we will take the help of DI.

If I am container I will inject you dependencies or resource base upon you requirements. So you have to pull it according to your requirement use it and leave it. Life of the resource maintain by container.

In spring framework we can achieve DI using two ways

1. Constructor base
2. Setter base

Using XML or Annotation

To achieve DI using XML as well as annotation Spring Framework provide two pre-defined API

Both are interfaces. BeanFactory is super interface and ApplicationContext is sub interface.

BeanFactoy

ApplicationContext

Achieve DI using Xml with BeanFactory interface.

Singleton : it mean only one object created and more than one reference but memory only one.

By default spring framework any class object created as singleton object. but if you want to created each time new memory using xml file we have to set the property scope=”prototype”

Day 2

21-03-2022

Spring auto wired: Auto wired is a spring framework features which enables you to inject the complex object dependency implicitly.

If you want to do di for complex object we have to do explicitly using ref attribute.

Or else use auto wired features.

Autowired

byType : inside xml we have to keep only one address bean tag of that type.

byName :

if more than one bean address tag available then we have to use byName.

In byName option id name and address class reference name must be same.

DI Using Annotation

Spring framework provided lot of annotation

@Component : This annotation we use on POJO or JavaBean class

// <bean class="com.Employee"> </bean>

Above <bean> tag is equal to @component annotation

By default id is consider as classname in lower case (camelNaming rules).

If class contains one word then id must be lower case

If class contains more than one word from second word onwards first letter upper case

@Autowired annotation : This annotation we have to use on complex property.

By default these annotation are not enable to enable these annotation we have to use xml file or class with @Configuration annotation.

Spring framework with JDBC (DataSource Features)

Data source :

DriverManagerDataSource one of the pre-defined class provided by Spring framework which help to get the DataSource details.

DataSource is a source of data which provide database connection in high secure, loosely coupling and with singleton concept. Before Spring framework if we want to achieve data source features we were depends upon the Application server like Web Logic or JBoss. But Spring framework with help of few jar file we can achieve data source features very easily.

Spring framework provided pre-defined annotation ie @Repository. This annotation we have to use on that class jdbc or orm code.

Spring framework provided pre-defined annotation ie @Service. This annotation we have to use on service class.

22-03-2022

Spring MVC :

Spring MVC internally follow MVC architecture. Provided annotation for Model layer (Server layer, Dao layer) and Controller layer. And provide spring tags to improve view layer.

class MyServlet extends HttpServlet {

public void doGet() {}

public void doPost() {}

}

@Controller

class MyController {

@RequestMapping(value=”sayHello”,method=RequestedMethod.GET)

public ModelAndView sayHello() {

ModelAndView mav = new ModelAndView() ;

mav.setViewName(“display.jsp”);

}

@RequestMapping(value=”sayHi”,method=RequestedMethod.GET)

public ModelAndView sayHi() {

ModelAndView mav = new ModelAndView() ;

mav.setViewName(“info.jsp”);

}

@RequestMapping(value=”checkInfo”,method=RequestedMethod.GET)

public String sayHi() {

// do some logic.

return “display”;

}

}

@RequestMapping annotation is use to map the request url pattern and method.

In Spring MVC we can’t call controller mean those classes with @Controller annotation directly we have to configure FrontController in web.xml file. Spring provided pre-defined class ie DispatcherServlet is a pre-defined class it behave like a FrontController in Spring MVC which we have to configure in web.xml file.

If controller method return string. We have to configure view resolver in xml file and their view provide the details about view.

Spring MVC with Database (Jdbc or Hibernate )

SpringMVCWelcome

index.jsp (View )----------------- web.xml (in web.xml file we have to configure DispatcherServlet FrontController class) (DispatcherServlet receive any request come from view and load the Spring configuration file which start with prefix sevletname-servlet.xml (servlet name in servlet tag is A or dispatcher ie A-servlet.xml or dispatcher-servlet.xml)-dispatcher-servlet.xml (we enable @Controller annotation using <context:component-scan base-package="com"></context:component-scan> :--

LocalSessionFactoryBean is a pre defined class provided by Spring framework which help to configure Spring framework with Hibernate.

This class contains lot of prorates

dataSource -

annotatedClasses - Entity class with @Entity and @Id annotation

23-03-2022

Spring boot : Spring boot is a type of spring framework module which help to develop the application very fast. Spring boot provide RAD (Rapid Application Development) features to Spring framework.

Limitation of Spring Framework

1. Huge framework.
2. Multiple setup while doing different application.
3. Multiple configuration files means more than one xml file we have to create.

Spring boot is a basically in which bootstrap or quickly start up a spring application.

Spring boot = All spring modules (core, context, mvc, dao, orm, aop, security ) - no xml file + few annotation + embedded server ie tomcat or jetty.

Spring boot itself is core Java. In Spring boot we will write main class. But which help to create any type of application.

Spring boot component

1. Spring boot starter
2. Spring boot AutoConfigurator

Spring boot starter : Spring boot starter is one of the major key or component of Spring boot framework. The main responsibility of Spring boot starter it to combine a group of common or related dependencies (jar files) into a single dependencies jar files.

Spring boot starter

Base starter

Web starter

Jdbc starter

Jpa starter

Security starter

Testing starter

Starter simply provides all dependencies that are likely to need when we develop any type of specific application.

Spring boot AutoConfigurator

To develop Spring application we require lot of configuration details using XML of annotation.

The solution for above problem given by AutoConfigurator. The main responsibility of Spring boot AutoConfigurator is to reduce the spring configuration details. In spring boot no xml file and we are using few annotation.

@SpringBootApplication = @Configuration +@ComponentScan + @EnableAutoConfiguration

Spring boot non web application with DI

Spring boot web application :

Spring boot provided web starter to create the web application.

Spring boot web application web.xml file and spring configuration file not require.

Spring web starter internally provided tomcat server.

24-03-2022

In Spring boot if we want to provide spring boot configuration details like database details, server port number and more we have to create application.properites or application.yml

This file must be created insider a resource folder.

Pom.xml file is a part of maven tool it is not a part of Spring boot.

Spring boot with view as JSP and connecting database using JPA

JPA is a specification and Hibernate is a implementation

Spring framework with Hibernate

Spring boot with JPA

25-03-2022

Spring framework provided its own form tag which help to bind form property to JavaBean or entity class directly.

If we want to use spring form tag we have to take the help of talib tags to provide the spring form details like jstl.

Controller : Controller is intermediate between View and Model layer.

Controller  Normal Servlet class

Spring MVC : class with @Controller annotation

If Controller is Normal Servlet class or Spring MVC controller then view must be HTML and JSP.

Our View technologies tightly coupled with Java Controller.

req

Client HDFC

res

Java

Servlet/JSP

Spring MVC

Spring boot

req java(res)

Client Amazon XML/JSON Payment

res Spring boot (.net/php/python)Google pay Python

Paytm Java

Phone pay .net

Net banking php

Credit card

Java is pure object and platform independent programing language but language dependent.

Web Service : Giving the service for web application when both application running using same technologies or different technologies or language.

XML : eXtensible mark up language

JSON : Java Script Object Notation

Both are use to share the data between two technologies.

Java Object

Employee emp = new Employee();

emp.setId(100);

emp.setNam(“Ravi”);

emp.setSalary(12000);

xml

<Employee>

<Id>100</Id>

<Name>Raj</Name>

<Salary>12000</Salary>

</Employee>

JSON

{“id”:100,”name”:”Raj”,”salary”:12000}

Web Service :

Web service is platform, language and browser independent.

Two types of web service

1. SOAP base Web Service : Simple Object Access Protocol. SOAP web service is base upon SOA (Service Oriented Architecture).

In SOAP web service we can consume and produce data only in the form of XML. XML is very heavy. DTD and XSD.

1. REST Full Web Service Representation State Transfer. Expose our resources (Servlet, JSP and Spring MVC or Spring boot) as a web service.

Rest full web service is not a standard like SOAP. It is architecture style.

@Controller annotation is replace by @RestController.

All method which present in @RestController we can use to create REST API . Those Rest API use to create resource (post method), get resource (get method), delete resource (delete method) , update resource (update methods).

If we use @RestController any technologies can call our application

Ie Java, .net, php, python, angular or react js.

Rest Web Service is ready to consume and produce the data in any format on demand like xml, json, plain text, html etc.

HTML and JSP ----------------------- Angular Framework

Frontend technologies Backend technologies

Html,css,js, typescript Angular REST API Spring boot

Or Asp.net

React JS Php

Python

Spring boot with RestFull Web Service

Get method

Get resource

Select query

1. Plain text format
2. Xml format
3. Html format
4. Return one complex object in json format
5. Return more than one complex object in json format.

We can pass the value for get method using two ways

1. Query param

Single value : URL?key=value

Multiple value : URL?key=value&key=value

By default html form with Get method internally use query param concept.

1. Path param

Single value : URL/value1

Multiple value : URL/value1/value2

Post method

Create the resource

Insert query

Put method

Update the resource

Update query

Delete method

Delete the resource

Delete query

In spring boot we use @RequestBody annotation to receive the data from request body part in the form of JSON.

We can’t call Post, put and delete method through URL. We can call only get method.

If you want to check post, put and delete method we have to use some plugin in browser to verify the Rest Full web service method like put, delete and post.

Rest plugin for browser, ARC or Post man or SOAP UI etc.

Id,name,salary

If we are planning update name and salary using id property use put method. update whole existing object.

If we are planning update only one or partial object using id property using patch method. partial whole existing object.

Spring boot with JPA to do CRUD Operation, Create Employee, Delete Employee, Update Employee, Retrieve employee using different conditions.

Spring provide pre-defined official website which help to create the spring boot application very easily.

<https://start.spring.io/>

Maven project

Language  Java

Spring boot  2.5.11

Artifact Id : Project Name

Package name : com

Java Version 11

Add dependencies

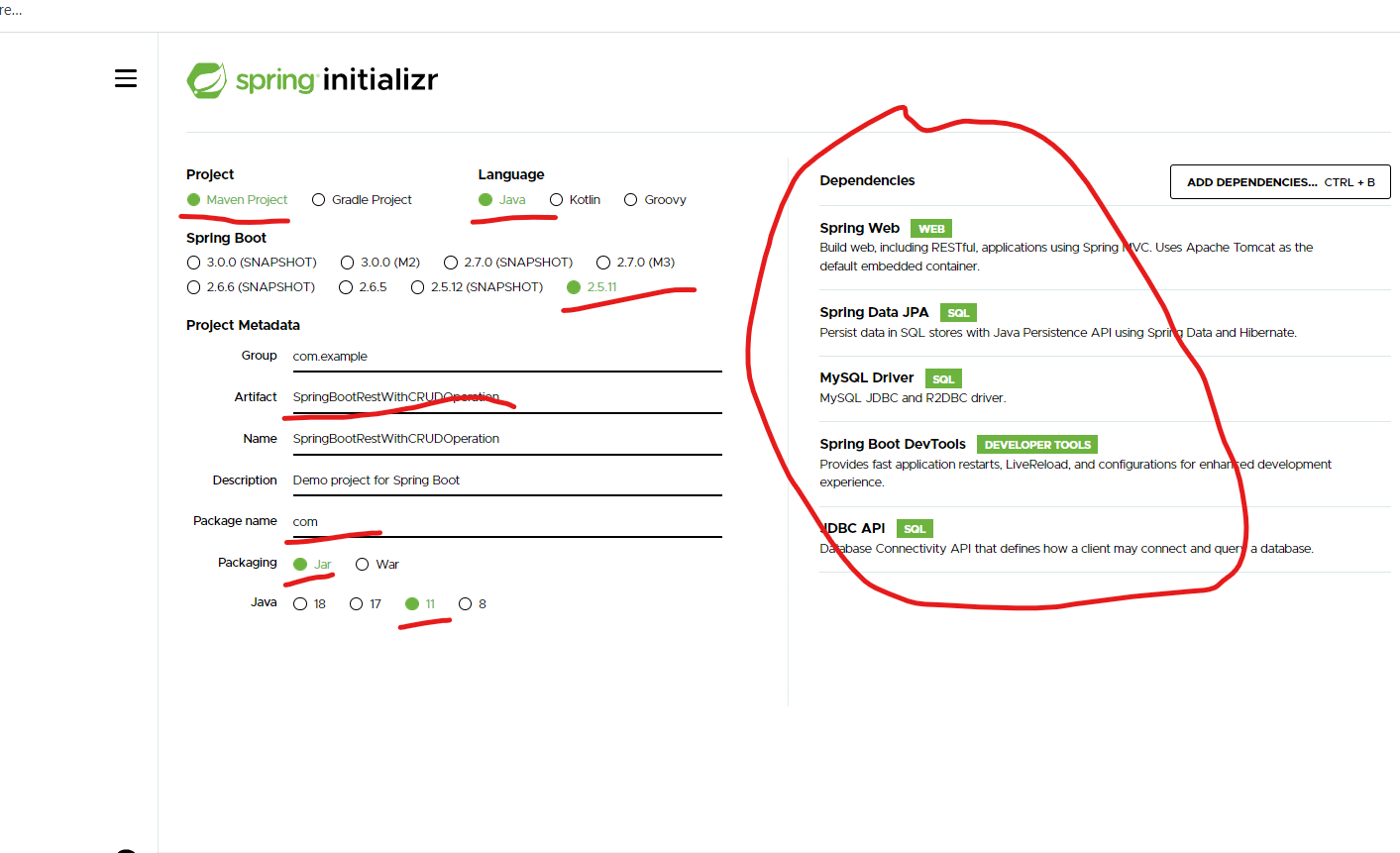
Web starter - Spring web starter

JDBC starter  JDBC provide data source features

JPA starter - to use ORM Features

My SQL Connector  To connect the database

Dev starter - whenever we do any changes in project it auto refresh the project.



This annotation we have to use map root path

@RequestMapping annotation is generic for all method. get, post, put and delete

These annotation we have to map specific http method.

@GetMapping is method specific for Get Method

@PostMapping is method specific for post method

@PutMapping is method specific for put method

@DeleteMapping is method specific delete method

29-03-2022

Monolithic Service :

Customer Module Controller, service, dao, entity -table

Login module Controller, service, dao, entity -table

Employee module Controller, service, dao, entity -table

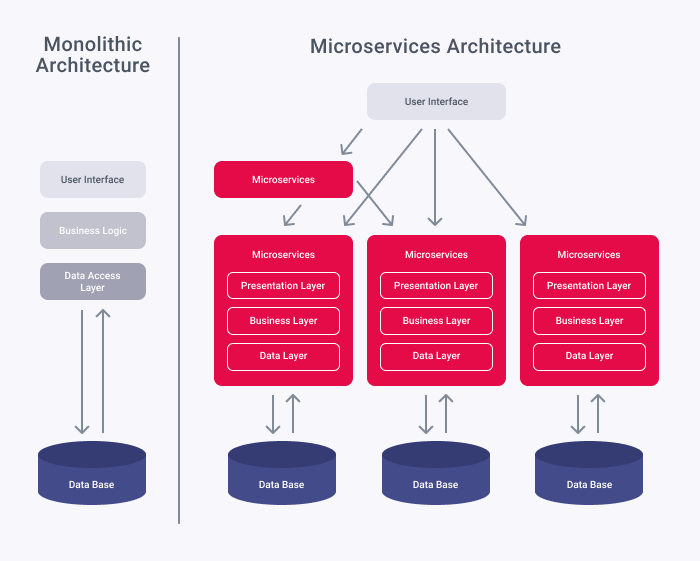
Feedback module Controller, service, dao, entity -table

Etc

Micro service :

We can create each module separately and deploy separately. Those module may be created using same technologies or different technologies using same database or different database may be in os or different os.

Micro service help in breaking the boundaries of large application and build logical independent smaller system or application.



In General, all micro service we can deploy separately and those services are communicate to each other using http and rest API call.

Using Spring boot we can create spring micro service application very easily.

To do this one Spring provide set of modules

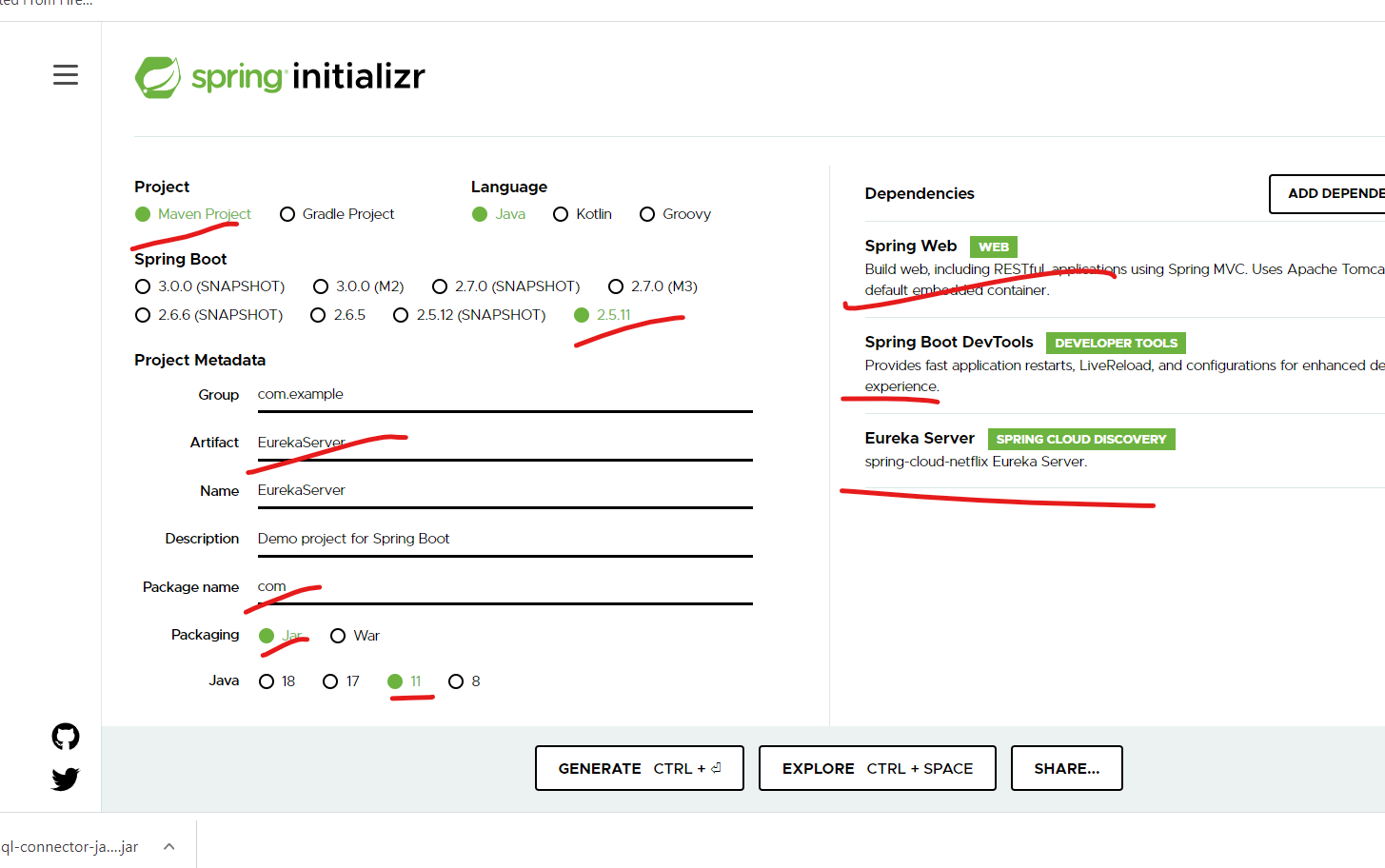
Spring cloud

Spring micro service

Spring cloud provide open source server ie Eureka Server to deploy more than one micro service.

We will create the Spring boot project to run the Eureka Server

EurekaServer



When we deploy any micro service project they search eureka server running on port number 8761.

Else every micro service project in application.properties file we have to write default about eureka server port number. It may be 8080.

By Spring every spring boot web project default port number 8080.

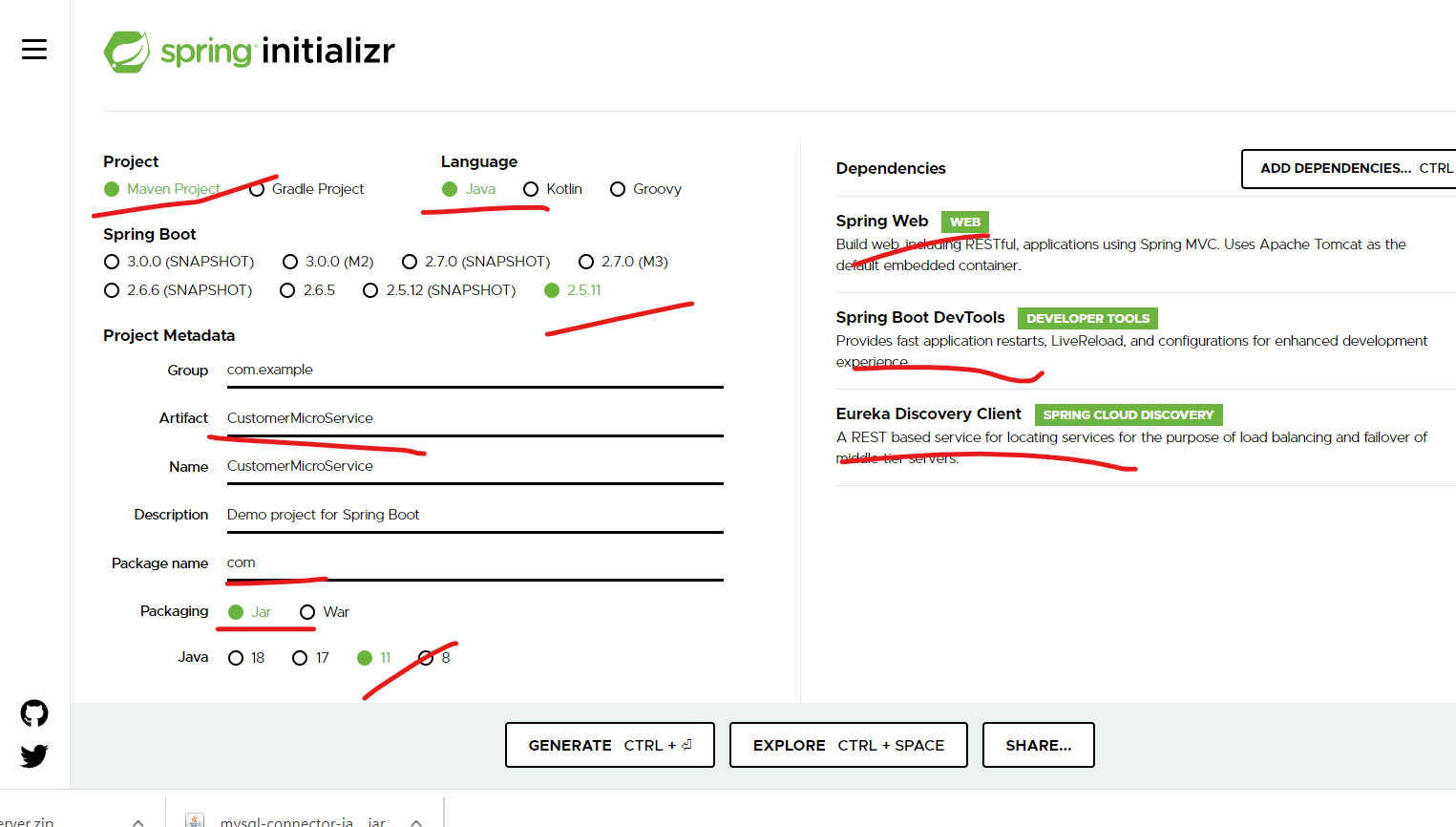
eureka.client.fetch-registry=false

eureka.client.register-with-eureka=false

We want this application as a server not a client

Now we will create micro service projects

CustomerService



Testing

Finding the defect or error or bugs the application.

Running the application itself is known as testing the application.

But if we develop layer architecture application like

Controller

Service

Dao

Bean

Resource

class Operation {

public int add(int a, int b) {

int sum = a+b;

return 0;

}

}

Testing divided into two types

1. Black box testing
2. White box testing

Black box testing doesn’t matter process or coding.

Input ----Process ----Output

White box testing

Input ----Process ----Output

Unit Testing : Unit testing is type of white box testing where individual unit of component of software or application are tested. Unit means where we write the code like function or method or class.

jUnit is open source framework which provided set of API which help to do the testing for Java application.

Junit 3.x without annotation

Junit 4.x with annotation

Junit 5.x more annotation with java8 features.

In unit testing we have to use test case and test suite

jUnit 4.x and 5.x provided pre-defined annotation @Test. Which help to do the testing for java application. Test case is use to write more than one test function which help to do the testing for functional functionality.

Test suite is to run more than one test case.

Every test case contains more than one method with @Test annotation