Java Phase 3 :

11 classes

Day 1

16-08-2022 Spring Framework and spring boot

JEE : Servlet, JSP and EJB.

View -🡪 HTML and JSP.

Controller -🡪 Servlet

Model -🡪 JavaBean or Entity class

Service class

Dao class

Web Container is a part of server responsible to take care execution of servlet and jsp. Web container will create the object of servlet and jsp and call life cycle methods.

EJB : Enterprise Java Bean : EJB container which is part of application server is responsible to improve the model layer. EJB help us to improve the model layer.

View 🡪HTML or JSP

Controller -- > Servlet

Model 🡪 EJB

EJB is very heavy component as well as very complex to run the application. To run the ejb application we required application server.

Solution for EJB application spring framework.

EJB Vs spring framework.

If we develop application with the help of servlet and jsp. It is not mandatory we have follow MVC design pattern or architecture.

Design pattern : Best practice or solution for repeating problem.

27 design pattern GOF

Creational design pattern

Structural design pattern

Behavior design pattern

Framework : Framework contains lot of pre-defined classes and interface which internally connected to each other to perform specific task. Framework also known as prototype or template but not a final product. If we develop any application using any framework 70 to 80% task taken care by framework. Framework internally allow all design pattern. The implementation of design pattern is taken care by framework.

Struts : struts is open source web framework provided by Apache. Struts internally follow MVC design pattern. It provided lot of classes to improve controller and model layer. Struts internally provided FrontController design pattern.it provide whole application flow. Struts internally provided pre-defined FrontController class.

Struts known as Controller centric framework.

JSF : Java Server Faces : JSF is open source web framework provided by Oracle. JSF internally follow MVC design pattern. It provided lot of classes to improve controller and model layer. JSF internally provided FrontController design pattern.it provide whole application flow. JSF internally provided pre-defined FrontController class.

JSF is known as View Centric Framework.

JSP -🡪 JSF VS Angular Framework or React JS

Hibernate : Hibernate is a ORM framework which help to improve DAO layer. Hibernate is a replacement of JDBC.

Spring framework : spring is a open source light weighted layer architecture framework. Spring also known as onion architecture framework.

Spring provided lot of pre-defined modules which help to improve all layer in the application. On demand we can use any module to improve that specific layer.

Spring core

Spring context

Spring web

Spring MVC : spring mvc it follow MVC architecture framework. Provided lot of classes and interfaces to improve controller and model layer. It provide FrontController design pattern.

Spring MVC is model centric framework.

EJB VS spring Framework

Spring DAO

Spring ORM

Spring AOP

Spring REST

Spring cloud

Spring security

Spring micro service

Spring expression

Spring integration etc

Spring Core and Context

IOC : Inversion of Control

DI : Dependency Injection

IOC : Inversion of control : IOC is a concept. It is programming design pattern. In place of creating or maintaining any resource explicitly allow to container create and maintain. If we create we are responsible to maintain the life of the resource. If container will create it will maintain properly. From container pull any resource whenever required use it and leave it.

DI : DI : Dependency Injection : DI is a implementation of IOC.

Using DI concept container inject the any external resource which we want. So we have to pull from container whenever we required.

Spring framework support two type of DI

1. Constructor base DI
2. Setter base DI

We have to configure constructor base as well as setter base DI using

1. Using XML file configuration
2. Using Annotation base configuration

We configure spring plugin in eclipse IDE

Constructor Base DI using XML file configuration

Now days we are developing spring framework application using two build tool

1. Maven
2. Gradle

Spring Provided to achieve DI for POJO class. Plain Old Java Object.

Spring container is a part of jar file. Which will help us to create the object for Plain Old Java Object. This is a normal class not to extends or implements Pre-defined class.

Day 2

17-08-2022 Spring Framework and spring boot

Constructor base DI using parameterized number of parameter and order of parameter must be match. We can’t partial DI using constructor base.

Setter base DI number of parameter and type of parameter not mandatory must be match. Using setter base DI we can achieve partial DI.

Spring framework do inject ie DI for primitive property implicitly

Autowired : autowired is a spring framework features which help to inject ie DI for complex property Implicitly rather than explicitly using property ref or constructor-arg ref attribute.

If we use auto wired features spring framework automatically scan for that complex object in xml file.

To Enable auto wired features using xml we have to use an attribute as autowire.

1. byType : in xml it must be contains only one bean declaration for that type of class.
2. byName : if we want more than one then we have to use byName. In byName option id name and reference name must be match.

Constructor and setter base di using Annotation

@component : This annotation we can write on class level. This annotation we will write on POJO or javaBean class. By default id name is classname in camel naming rules. If class contains one world in id must lower case. If class contains more than one word then second word onward first letter upper case.

@Autowired : This annotation we can use on complex property level, constructor level or setter method level.

By default @Component annotation is not enable. We can enable using

1. using xml file
2. using configuration class with few more annotation.

Using xml file.

Using configuration file

@Value : This annotation we have to use to property level to set default value.

Using java configuration class

@Scope : this annotation we can write on class level to make singleton or protocol. By default singleton.

Spring MVC : Model View Controller :

Spring MVC internally follow MVC architecture framework. It provided front controller pre-defined class ie DispatcherServlet. Which we have to configure in web.xml file.



@Controller : this class now behave like Servlet.

class MyController {

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

public ModelAndView sayHello() {

ModelAndView mav = new ModelAndView();

mav.setViewName(“display.jsp”); /RequestDispatcher

return mav;

}

}

Day 3

22-08-2022 Spring Framework and spring boot

First we have to create dynamic web project with tomcat server

Then we have to convert this project into maven project . right click on project and select the option as configure and then convert to maven.

Then open the pom.xml file and which java version in release tag may sure 1.8 or 11 or 15 etc.

Then we have to add the spring mvc dependencies.

Then in web.xml file add the dispatcher servlet as front controller.

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

url pattern is /. It means url can be hello, hi etc.

Then create spring configuration file start with pre-fix servlet name present inside <servlet-name> tag followed by servlet.xml ie dispatcher-servlet.xml

And while creating file please select the option as context.

dispatcher-servlet.xml

<context:component-scan base-package=*"com"*></context:component-scan>

It is use to enable @Controller annotation.

Then we have to make normal java class with Annotation @Controller and inside that class we can write more than one method with return type is ModelAndView. On those method we have to write @RequestMapping annotation with path and method options.

package com.controller;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

@Controller // this class like servlet.

public class MyController {

// doGet method with meaningful method name

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

public ModelAndView sayHello() {

ModelAndView mav = new ModelAndView();

System.out.println("I Came here");

mav.setViewName("display.jsp"); // RD forward

return mav;

}

}

Flow of the application

Index.html --🡪 web.xml 🡪dispatcher-servlet.xml -🡪MyController 🡪 sayHello with path inside @RequestMapping annotation -🡪 Display.jsp

@Repository : This annotation we have to write on dao layer.

Dao layer code written using jdbc or hibernate or jpa etc.

@Service : This annotation we have to write on service layer.

DataSource : DataSource provide source of database connectivity. Before Spring framework if we want to achieve DataSource Features we were depends upon application server. But After spring framework we have to add spring jdbc dependencies and which provide Data Source features with application server.

If we want to achieve data source features with spring framework we have to add spring jdbc dependencies.

To get the Data Source features we have to configure in spring configuration file.

Spring JDBC with DAO : Spring dao (jdbc) provided pre-defined API it JdbcTemplate. JdbcTemplate wrap core jdbc code and provided lot of pre-defined method to do the task very easily in dao layer.

Statement Vs PreparedStatement Vs CallableStatement.

23-08-2022

Spring MVC with Hibernate (DAO layer using ORM (Hibernate))

Hibernate.cfg.xml

Configuration con = new Configuration();

con.configure(“hibernate.cfg.xml”);

Hibernate doesn’t provide any features of DI and IOC.

This features provided by Spring framework.

Spring framework doesn’t provide any features like a ORM

It will provide by ORM tool like Hibernate or JPA(Java Persistence API)

Spring framework provided pre-defined class.

LocalSessionFactoryBean which help to configure spring framework with orm tool like hibernate.

This class is part of Spring orm dependencies.

*LocalSessionFactoryBean this class provide as SessionFactory interface DI.*

Spring MVC with ViewResolver :

In Spring MVC application we can write more than one method inside a controller class with return type as a String rather than ModelAndView class.

If we use return type as a string then in spring configuration file we have to configure ViewResolver with pre-fix and suffix.

ViewResovler we can configure using Java class or inside a spring configuration file.

Spring Boot :

Spring boot is another module from spring framework. Which help to develop the application using RAD (Rapid Application Development).

Spring boot is a bootstrap for all spring modules.

Limitation of spring framework

1. Spring framework is very huge framework.
2. We can configure using xml file or annotation.
3. We have configure all spring pre-defined classes using annotation or xml file Like DisptcherServlet, ViewResolver etc.
4. Depending upon the application we have to add more dependencies.
5. If we add Spring MVC with Hibernate.
6. XML file configuration and Dependencies download.

Spring boot = All Spring modules – No XML file + Few annotation + embedded Web Server.

Spring boot itself is a core java or standalone project which help to create any type of projects.

In Spring boot we are going to use main method. This main method available in development mode not in production mode.

Spring boot components

Spring boot starter : It combines more than one jar files or dependencies on one dependencies as starter based upon type of projects.

Ex : Web Starter

Jdbc starter

Jpa Starter

Testing starter

Security starter

Logger starter

Etc

Spring boot auto configuration :Spring boot auto configuration helps us automatically configure a spring application based on the dependencies added in the form of starter in the classpath.

This features remove all xml configuration and it provided few annotation do the configuration based upon the all jar files present in classpath.

@SpringBootApplication = @Configuration + @ComponentScan + @AutoConfiguration

This annotation we have to use on class level which contains main methods.

Spring boot with DI

Spring boot using two build tool

1. Using Maven then we have use pom.xml. pom.xml file is part of maven tool
2. Using Gradle : xml less build tool. Build file present to write configuration.

Spring Boot with MVC and View as a JSP

In Spring boot we use two types of file to write all configuration details like port number change, database details, security information, spring micro service details.

1. properties file application.properites
2. yml file application.yml

25-08-2022

@SpringBootApplication

This annotation by default scan current package and sub package of current package of main method.

But if controller, service class, dao class and java bean class part of different package then we have to scan.

@SpringBootApplication(scanBasePackages = "com")

Spring boot with Database we can use JPA (Java persistence API) or Spring Data.

Spring boot doesn’t support Hibernate API it support JPA.

JPA Vs Hibernate

JPA is a specification and Hibernate a implementation of base upon a JPA.

Hibernate is a framework part of jboss.

JPA is a type of EJB.

SessionFactory EntityManagerFactory

Session EntityManager

Transaction Transaction

Save persist

Delete remove

Update merge

Get find

HQL JPQL

**Make an E-commerce Website for Sporty Shoes .**

We required two user

Admin and customer/normal user

No signup for admin user only signin

We required signin and signup for customer or normal user.

After admin done the login successfully we have to admin dashboard.

1. Add product pid,name,price,url,category, quantity etc. category base upon typeofshoe or company.
2. Update the price and quantity.
3. Delete product using pid.
4. View all products.
5. View all user details.
6. Search user by emailid or phonenumber.
7. Admin can view all orders details
8. Filter by Date and category.

Once user created the account after user can login to application.

User Dashboard

1. View all product.
2. User can place the order.

Oid, username, productname, category, price, quantity, doo.

You have to decrease the number of quantity from product table.

1. User can view its own order details.

Phase 3 end project you have to do with spring mvc with Hibernate and View as JSP.

Web Service :

Req

Client HDFC XML/JSON HSBC

res

Spring or Servlet asp.net

Or JSP Python

Php

Amazon payment

Spring boot credit card

Debit card

Phone pay

Google pay

Paytm

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

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

XML :eXtensible markup language

JSON : JavaScript Object Notation

These are two format of data which help to share the data between two technologies.

2 types of web service.

1. SOAP Base Web Service or Big web service : Simple Object Access Protocol This web service base upon SOA (Service Oriented architecture) .

In SOAP web service we can consume and produce the data only in the form of XML.

1. RESTFull Web Service : Representational State Transfer : in RESTfull web service we can consume and produce the in any format on demand like html, text format, json,xml etc.

In Restfull web service we expose our resource like Employee, customer, manager, order, account as Web service in the form of xml or json on demand.

If spring boot or spring MVC controller is normal controller with @Controller then view must be JSP or HTML.

JSP and HTML code tightly coupled with controller ie Java.

If we make spring boot or Spring MC controller as rest full controller with the help of @RestController then view can be any technologies like ASp.net, PHp, python, Angular or React or any Rest Client application can call our application.

With the help of @Restcontroller annotation we will create the REST API. It will provide us URL those url are use to store, delete, update and retrieve data from database using jdbc or hibernate or jpa in the form of json or xml.

According to RestFull Web service. http five methods we have to use

Resource means entity or table Employee, customer, Manager, Logic, Product, Order etc.

Get : Get the resource

: Get all employee details

: Get specific employee details using some property can be id, age, name etc.

Select query

Select query with where clause

We can pass the value to check the conditions.

Using two technique

1. Query param
2. Path param

Post : Create the resource

Insert the record in database like Employee, customer, manager, order, product etc.

Put : It is use to update the resource

Update query

It is use to update whole resource.

Patch : it is use to update the resource

Update query

Patch is use to update partial property.

Delete : delete the resource

Delete query :

Delete record using id or any other property.

Swagger UI : Creating the Documentation for REST API.

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

Addition of two numbers

Read a,b,

Sum = a+b;

Write 0;

Testing means we are running the program without main method or function to check function functionality.

2 types of testing

Black box testing

Input Process Output

White box testing

Input Process Output

Manual testing

Automatization testing

Unit testing : Unit testing is a type of white box testing where individual unit of component of software are tested.

Unit means smallest set of code written in method or functions.

jUnit is a open source light weighted framework provided by third party which help to do the testing for java code.

jUnit 3.x without annotation

jUnit 4.x with annotation

jUnit 5.x with few more new annotation with combination of more than one framework.

Test case : it is a type class which contains more than one test method with annotation @Test which help to test function functionality .

Test suite : To run more than one test case classes at the same time when we can use test suite. Test suite is a combination of more than one test case classes.

Each test case class contains more than test method.

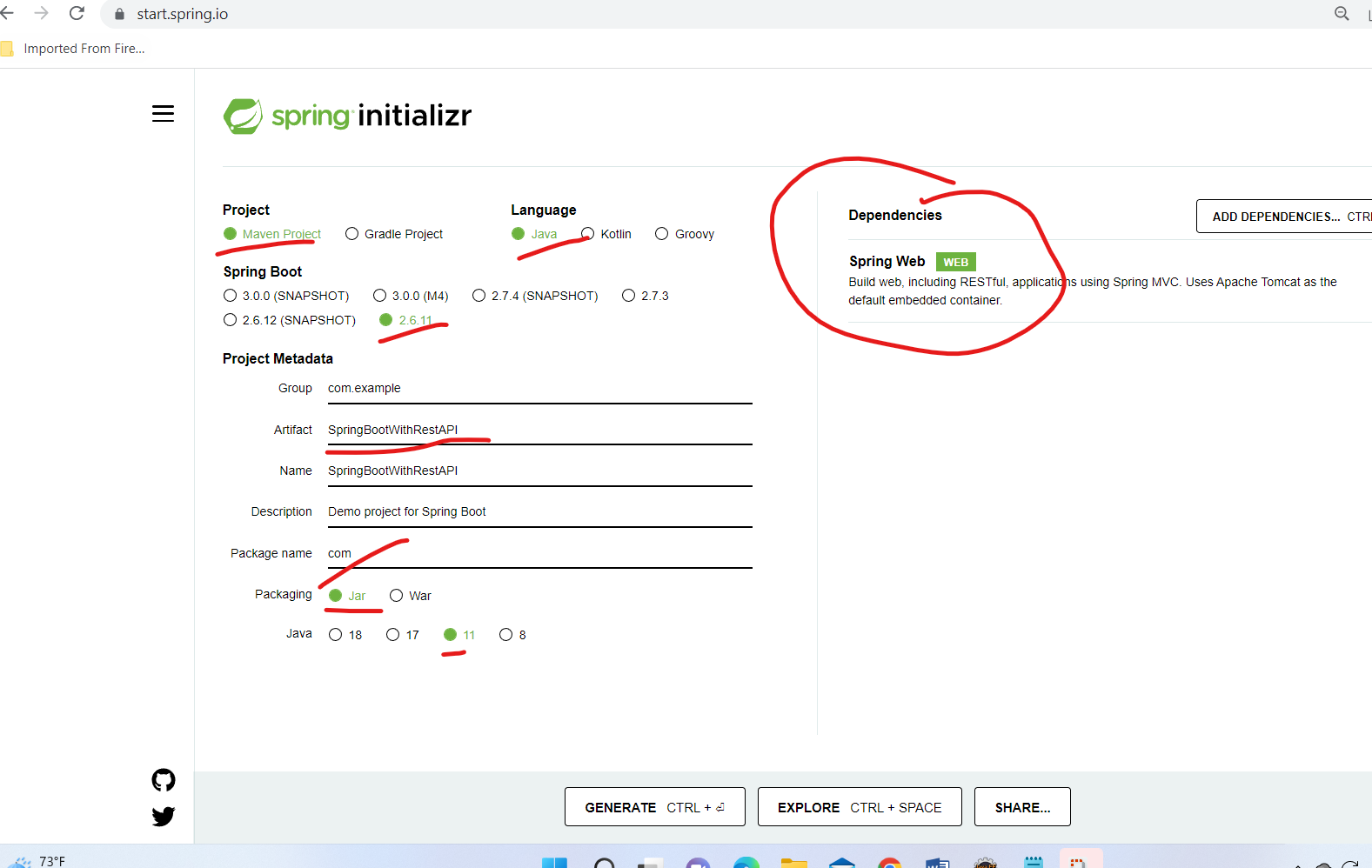
Assertion : junit provided set of assertion api which contains set method which help to check actual and expected output.

29-08-2022

Spring micro service

Sprint boot official website which help to create the project very easily.

<https://start.spring.io>



Micro Service

Small service which is responsible to do the small task independently ie creating, developing and deploying independently.

Monolithic service : This type of service is responsible to do all task in one project base upon project requirements.

Monolithic service Vs Micro Service

While developing big application we required lot of modules

Like login module, customer module , manager module, account module, loan module, insurance module etc.

According to micro service we can create, develop and deploy the application with different database according to our requirements in any language.

Login module using Java with database mysql : small service running on port number 8181

Customer module using java with database oracle : small service running on port number 8282

Account module using python with mongo db : small service running on port number 8383

These all micro service interact with each other using rest api.



Using spring boot we can develop the micro service application very easily.

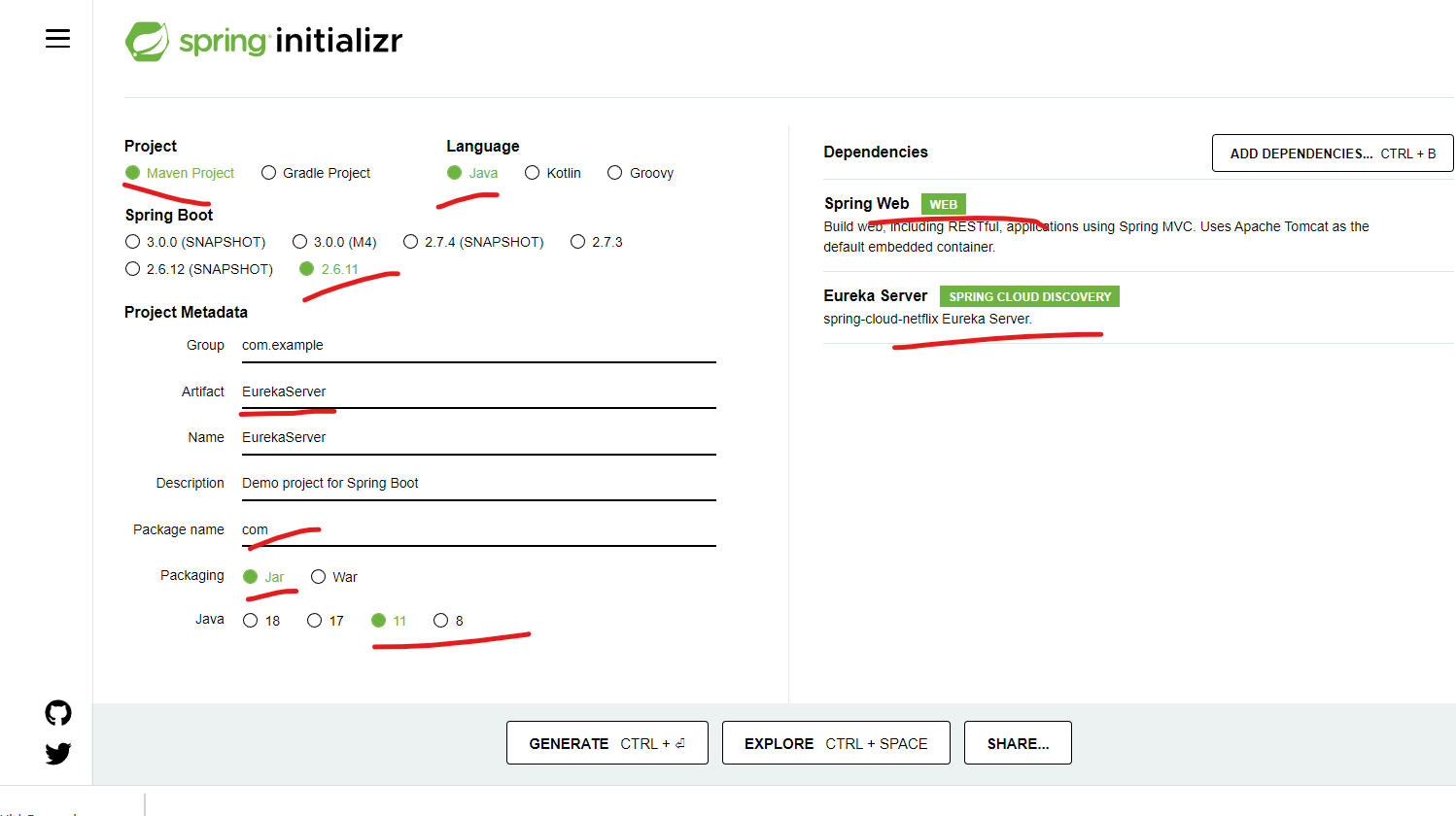
Spring boot provided one of the open source server ie Eureka Server. This server is responsible to hold all micro services develop using spring boot with java technologies. Eureka is a type of open source server.

So we will create micro service projects.

First project : Eureka Server

Two starter : Web

Eureka



We have to run the eureka server on port number 8761. Default port number of Eureka Server is 8761. Whenever we run micro service client project with eureka client by default it will search eureka server port number as 8761,

Creating first micro service client application

FirstMicroServiceClient

Starter Web

Eureka Client

SecondMicroServiceClient

Starter Web

Eureka Client

Creating third micro service interacting with database with spring data.

In Spring Data we no need to write any logic in dao layer. We have to make normal interface and that interface must be extends JpaRepository interface. And JpaRepository is a pre-defined interface which provide lot of pre-defined DAO or ORM method with default implantation with Java8 features.

So we have to make only normal interface with Anntation @Repsitory and that interface must be extends JapRespository and call all pre-defined method inside service class.

Spring data completely abstracted all dao methods.

Third micro service EmployeeManagementSystem with Spring data

EmsMicroService

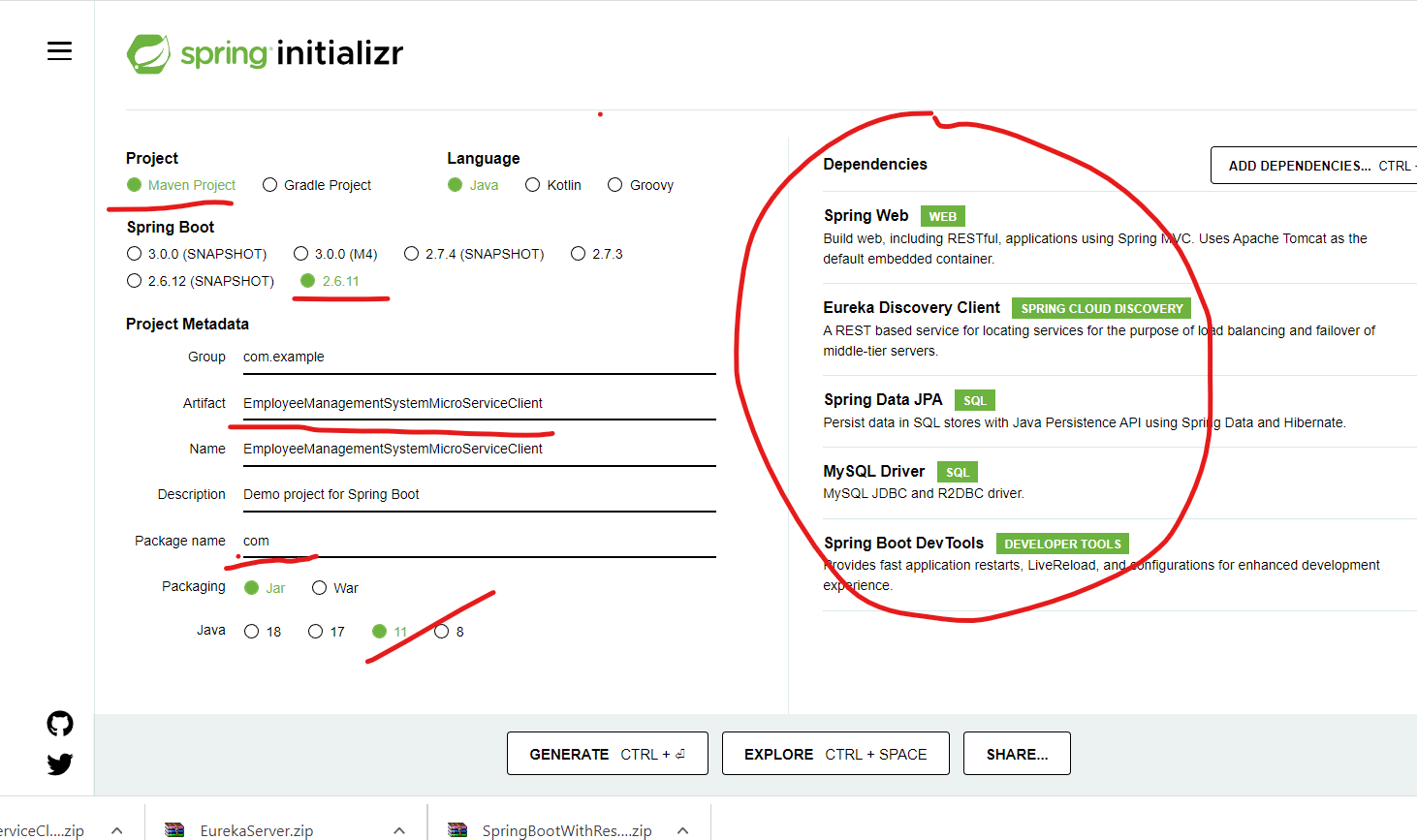
Starter 🡪 Web starter

Jpa starter

Mysql connector 🡪 dependencies

Eureka client starter

Devtool : it will provide a features to auto refresh the application automatically when we save the program.



One micro service want to communicate with another micro service for retrieve, store, update and delete etc.

Employee Management System micro service provided five rest method to store, delete, update, retrieve, retrieve by id etc.

Manager micro service :

Phase 3 :

Online Examination We have to create Set of REST API

For Admin and Student (Participants) REST API.

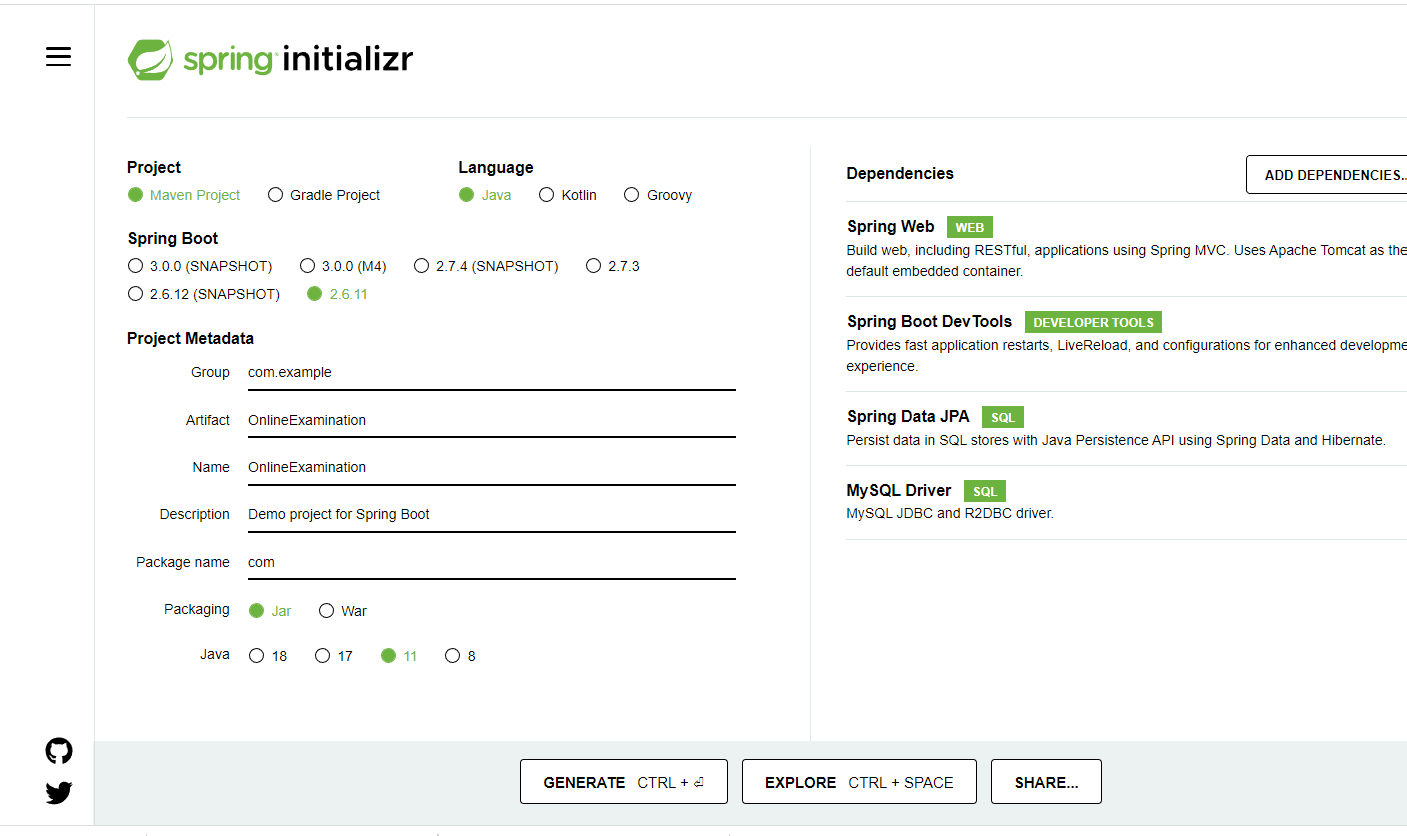


Table --🡪 login -🡪 emailed 🡪 PK

Password

Typeofuser

create table login(

emailed varchar(25) primary key,

Password varchar(25),

Typeofuser enum('admin','participants'));

No signup for admin user only we have to create signin for admin user

Take example username for admin is [admin@gmail.com](mailto:admin@gmail.com) and password is admin@123

<http://localhost:8080/admin/signin>

if username and password correct return successfully done else failure.

<http://localhost:8080/admin/changePassword>

using emaild and new password

new password not equal to old password.

Signup and signin for participants user.

<http://localhost:8080/admin/AddQuestions> Add question details

question table 🡪

qid -🡪 PK with auto-increment

question

op1

op2

op3

op4

CorrectAns

<http://localhost:8080/admin/findQuestionById/1>

<http://localhost:8080/admin/findQuestionById/2>

return question details in json format.

Quiz table 🡪

Qzid 🡪PK

Title 🡪

Subject/Category

Qid 🡪 FK

<http://localhost:8080/admin/createQuize>

we have to pass title, Subject/category, Qid

<http://localhost:8080/admin/checkUserResult>

username/emailid, score , title, standing position

raj, 10, Maths, 1

<http://localhost:8080/participants/signup>

first we have to store the record in login table with emailid,password and typeofuser(roles) is participants.

Then insert the record in participants table.

participants 🡪 table

pi dint auto-increment

fname

lname

age

phonumber

emailid

create table participants(

pid int primary key auto\_increment,

fname varchar(25),

lname varchar(24),

age int,

emailid varchar(25),

password varchar(25),

foreign key(emailid) references login(emailid));

<http://localhost:8080/participants/signin>

checking username and password from database if correct return successfully login else return fail.

User quiz

User\_quiz table 🡪

Uqid 🡪Pk auto increment

Emailid 🡪FK

Qzid-🡪 FK

QId🡪FK

UserCorrectAns🡪

1, [raj@gmail.com](mailto:raj@gmail.com), 11, 1111, A

<http://localhost:8080/participants/takeQuiz>

user check the result

<http://localhsot:8080/participants/checkResult>

emaildId and title/subject/qzid

Return result in string your score

<http://localhsot8080/participants/checkAllQuizDetails>

send req --🡪 with username and password and roles

check their token

Client Server

JWT

JSON Web token

123abc

Authentication and Authorization

Authentication is use to check our login details and base upon username/emailid/id and password it will allow to access the resources.

Authorization : it will check our role and base upon roles it will allow to access the resources.

If we are normal user we can do login to our laptop to use it but as a normal user roles we can’t install any software in C drive.

If we are admin user use can access all resource base upon our role.

We created 10 rest API

Those rest api we allow to access for authenticate user.

Among that 10 REST API only 6 rest api can be access by normal user and all rest api can be access by admin user.