Phase 3 : Spring Framework and Spring boot

Collection Framework

Framework : It provide set of API (Application Programming interface) which help to do some specific task. Framework is like a protocol or template but not a final product. If we develop any application using framework 70 to 80 task taken care by framework. If we develop any application using framework by default we are follow standard rules. The implementation of design pattern is taken care by framework. Design pattern mean best practise or solution for repeating problems.

Collection framework is a like Data structure.

MVC :

View -🡪 JSP/HTML

Controller -🡪 Servlet

Model -🡪 JavaBean, Service class, Dao class, resource (may using normal class java or xml file).

doGet and doPost : Login or Product object we are creating, Service class object we are creating, dao and resource layer we are maintaining.

EJB : Enterprise Java Bean : EJB mainly use to improve the model layer.

To run the EJB application we require EJB container.

EJB container is a part of Application Server. Application server heavy weighted server.

JPA (Java Persistence API).

JPA is type of EJB and it is a technologies. JPA is a specification. Hibernate is a framework and hibernate is a implementation of JPA.

**Web Server**  : Tomcat : it is a type of web server which contains only one type of container ie web container. Web container is responsible to execute servlet and JSP.

Application Server. Application server contains web container, ejb container, jms(java messaging service) container and more. It provide extra services like connection pooling, thread management, resource management, security etc.

In application server we can run servlet, jsp and ejb application.

Web Logic, JBoss, Web Sphere, Glashfish etc.

MVC

JEE

VIEW 🡪 HTML/JSP

Controller -🡪 Servlet

Model -🡪 EJB

Struts : Struts is a MVC base web base open source framework. Which help to develop web application. Struts internally follow MVC architecture. Struts is known as controller centric framework. Struts is a part of apache.

JSF Java Server Faces. JSF is a MVC base open source framework. Which help to develop web application. JSF also follow MVC architecture. JSF is known as View centric framework.

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

Spring modules

1. Spring core
2. Spring context
3. Spring Web
4. Spring MVC : Spring MVC is model centric framework.
5. Spring REST
6. Spring DAO
7. Spring ORM
8. Spring AOP
9. Spring security
10. Spring cloud
11. Spring boot
12. Spring micro service

Etc

void a() {

}

void b() {

}

void c() {

}

void main() {

a();

//b();

//c();

}

IOC and DI

IOC: Inversion of control:

IOC is a design pattern or programming pattern.

IOC is a known concept. In place of creating any resources or objects explicitly allow to create by container. If container will create it will maintain properly. Whenever we require we have to pull the resource from a container use it and leave it.

Using IOC we can achieve loosely coupling concept.

DI: Dependency Injection:

DI is a implementation of IOC. We can achieve DI

Using 3 ways

1. Constructor base DI
2. Setter base DI
3. Interface base DI

But support two types of DI constructor base and setter base DI.

Using DI we pull the resource from a container.

object : it is a concept

class : class is a blue print or template or user defined data type to create the object.

Class Employee {

}

Employee emp = new Employee();

emp is a object

In Spring framework container is part of jar files and xml files/annotation.

So if we want to achieve DI in Spring framework we have to configure using xml or annotation.

STS

DI Using constructor base with XML

Spring core dependency provide set of API which help to pull the object from a container.

<bean class=*"com.Employee"* id=*"emp1"*></bean>

Container will create the object and provide reference id for that object ie emp1.

ClassPathXmlApplicationContext is pre-defined class which help to load the xml file and reference of this class is ApplicationContext interface reference.

Auto wired : Auto wired features help in spring framework to enable complex property or objects to inject implicitly or it is a replacement of ref property for the complex objects.

DI using annotation

@Component : This annotation is class base annotation. This annotation we have to use on POJO class or generic annotation for normal Java class.

@Autowired : This annotation we can use on property level and that property must be complex.

By default @Component annotation not enable so we have to enable using xml file or using configuration class with some annotation.

DI using annotation with configuration class.

@Component

@Autowired

@Value

@Configuration

@ComponentScan

DataSource : It is use to provide the database connection with proper configuration.

Before Spring framework if we want to achieve Data source features it require mandatory application server. JNDI(Java Naming directive interface) look we have achieve data source features.

Spring DAO :

Spring DAO layer provide improvement on JDBC codding. Spring DAO provide set of API using jdbcTemplate.

JdbcTemplate api wrap the jdbc code and provide some extra functional to provide JDBC logic.

Spring ORM : Object Relation Mapping

Spring framework doesn’t provide any ORM tools it help interact existing ORM tools ie Hibernate, JPA, iBaties etc.

create table trainer(tid int primary key,

tname varchar(10),

tech varchar(10));

create table students(sid int,

sname varchar(10),

age int,

tsid int,

constraint ts\_fk foreign key(tsid) references trainer(tid));

Spring MVC : Spring MVC internally follow MVC architecture framework. It provided lot annotation to improve view, controller and model layer.

Spring MVC follow FrontController design pattern concept.

LoginController

AccountController

EmployeeController

StudentController

FrontController : it is a type of servlet class responsible to navigate sub controller depending upon the path.

Spring MVC provided pre-defined API ie DispatcherServlet. This class is a FrontController in Spring MVC.

We have to pass the request to Spring MVC application through html or JSP.

DispatcherServlet FrontController we have to configure in web.xml file or using java classes.

So old version spring MVC handler mapping we were configure in spring configuration file(xml) now replace by annotation that is @ReqeustMapping. This annotation help to map the request. Then specific controller will execute. Old version Spring MVC we were making Normal class and extends or implements Pre-defined controller API provided by Spring MVC. New Version Spring MVC all classes replaced by @Controller annotation. Inside that class we can write user-defined method and with the help of @RequestMapping we can make method may be get or post. @RequestMapping annotation provided pre-defined attribute ie method. Then controller class create do the AutoWired for service layer. What sevice layer return that output provide the DispatherServlet. In Spring MVC all method return type must be ModelAndView. That class help to details about Model and View layer.

Then dispatcherServlet class interact with ViewResolver class to display same output in different format. This class is optional. After conversation then DispatcherServlet send the output to view.



Once request receive from view. DispatcherServlet configure in web.xml file search spring configuration file with name servlet-servlet.xml

Ex : dispathcer-servlet.xml

Abc-servlet.xml

Inside EmployeeController we can do the Autowired for EmployeeService . Inside a Service class we can do the autowired for EmployeeDao class.

Inside EmployeeDao class we can do the Autowired for DataSource or SessionFactory

@Service

class EmployeeService {

}

@Repository

class EmployeeDao {

}

If we want to improve our View layer which was created using html / jsp with or with JSTL. The view technologies tightly coupled with controller layer.

Req

Client HDFC SBI

Res Java Asp.net

Spring MVC Php

Python

Java (Spring MVC)

Amazon

Google pay Spring MVC

Paytm Asp.net

Phone pay Php

Net banking Python

Credit card etc

Amazon pay

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

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

Command language to share the data is XML and JSON

**XML :** eXtensible mark up language

**JSON** : JavaScript Object notation

These two format will help to share the data between two technologies.

1. SOAP Base Web Service

Simple Object Access Protocol. SOAP Web Service is known as big web service. SOAP Web Service is base upon SOA (Service Oriented Architecture).

SB – Service Broker

Register the service

Search in broker Details in Broker

SR : Service Requester SP

Or -----SOAP Req -🡪

🡨----SOAP Res----

SC : Service consumer Service Provider

SOAP web service is one of implementation of SOA.

SB

They register their

Service details usig WSDL

They download SP

WSDL file

Method

SR SOAP REq / SOAP RES Asp.net

Java Accno and

Only in xml form Get balance

WSDL : Web Service Description language: This file contains service details like service name, type of parameter takes, return value, running port number with host details. It is a type of XML file.

SR : they download the WSDL file and using some tool or language they convert WSDL to respective language code which help to call the Service. With help of those classes we can send the request to service provide using SOAP req and SOAP res.

In SOAP Web Service we can consume and product the data only in the form of XML.

XML is heavy because to validate XML we have to use DTD or XSD.

SOAP is standard web service.

1. REST Full Web Service

Representational State Transfer : Using REST full web service we are making our server side resource (Servlet or JSP or Spring MVC Controller) as a web service.

Using REST web service we can consume and produce the data in any format base upon the client requirements.

XML or JSON or Text or HTML or Media type etc.

In REST Full web service we can not downloading any code. Using URL we send request and get the response.

REST Web Service support all HTTP protocol methods.

Rest web service is light weighted because JSON.

If my controller is normal controller or Spring MVC controller then view must be html or jsp. It tightly coupled with controller.

If we make controller as RestFull controller

Then view technologies can be

1. Asp.net
2. Php
3. Python
4. Angular
5. React JS
6. Any REST Client application
7. JSP view also possible.

Sprig boot : Spring boot is basically bootstrap or quickly start up an all spring modules.

Using Spring boot we can develop RAD (Rapid Application Development).

Limitation of Spring Framework.

1. Huge framework.
2. Multiple setup
3. Multiple configuration files.
4. Maintain the jar files.

Spring boot itself is a normal Java program which help to develop any type of spring application.

Spring boot = Spring all modules – XML file + few annotation + in build web server ie tomcat or jetty

Spring boot with Maven (pom.xml file it is part of maven tool).

Spring boot with Gradle no xml file.

Spring boot components

1. Spring boot starter
2. Spring boot auto configurator

Spring boot starter : The main responsibility of Spring boot starter to combine all group of common or related dependencies into single dependencies.

Spring boot starter

Spring boot web starter

Spring boot jdbc starter

Spring boot jpa starter

Spring boot testing starter

Spring boot security starter

Etc

Spring boot auto configurator : The main responsibility of Auto configurator is to reduce the sprig configuration file and provided few annotations.

@SpringBootApplication = @Configuration + @ComponentScan + @EnableAutoConfiguration

This annotation we have to write in main method.

@SpringBootApplication

class Demo {

public static void main(String args[]) {

SpringApplication.run(Demo.class,args);

}

}

Controller layer,

Service layer

Dao layer

application.properties or application.yml

This file contains all configuration details which help to run the application.

This file contains all database connectivity + more

Spring boot application – Non Web Application

[**Spring Boot Starter**](https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter) **Please add this starter**

**Spring boot with Web Project with RestFull Web Service.**

**@Controller annotation then view must be HTML/JSP.**

**@RestController =@Controller + @ResponseBody**

**Every method return type must be @ResponseBody then view can be any technologies.**

**Rest Full Web Service produce the output in different format**

**Return type is String.**

1. **Text Format**
2. **Html format**
3. **Xml format**

**Return type may be user-defined object**

**Then media type must be xml or json**

**Return type may be collection of user-defined objects.**

**Then media type must be xml or json**

**JSON : JavaScript Object Notation : Display the data in the key-value format.**

**class Employee {**

**id,name,salary;**

**setter and getter methods**

**}**

**Employee emp = new Employee();**

**emp.setId(100);**

**emp.setName(“Raj”);**

**emp.setSalary(12000);**

**So we have to convert this object in xml file or json format.**

**XML Format**

**<Employee>**

**<id>100</id>**

**<name>Raj</name>**

**<salary>12000</salary>**

**</Employee>**

**JSON Format**

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

Passing the value for Rest full web service using get methods.

1. Using query param :

URL/path?key=value;

url/path?key=value&key=value;

this technique if form with get methods then internally they use query param concept.

1. Using Path param

URL/path/v1

URL/path/v1/v2/v3

According to REST API

1. Get method is use to get the resources. Resources Employee, customer, Manager, Order, Account, Login etc.

If we compare with database

Select query

Select all employees

Select specific employee based upon id ie path param or query param.

1. Post method is use to store the resources. Resources Employee, Customer, Manager, order, Account or Login etc.

Insert Query.

We can’t call post, put and delete method through browser.

To check these methods we have to use plugins.

Rest Client plugins

Postman client

Rest Client for chrome.

1. Put method : This method is use to update the Resources.

Customer, Login, Employee, Order etc.

Update Query

In post method we can pass partial object.

1. Delete methods : This method is use to delete the resource using some property. This value we have to pass through path params.

Spring boot with REST API and Database using JPA or Spring Data

Spring WEB starter

Spring JDBC starter

Spring JPA Stater

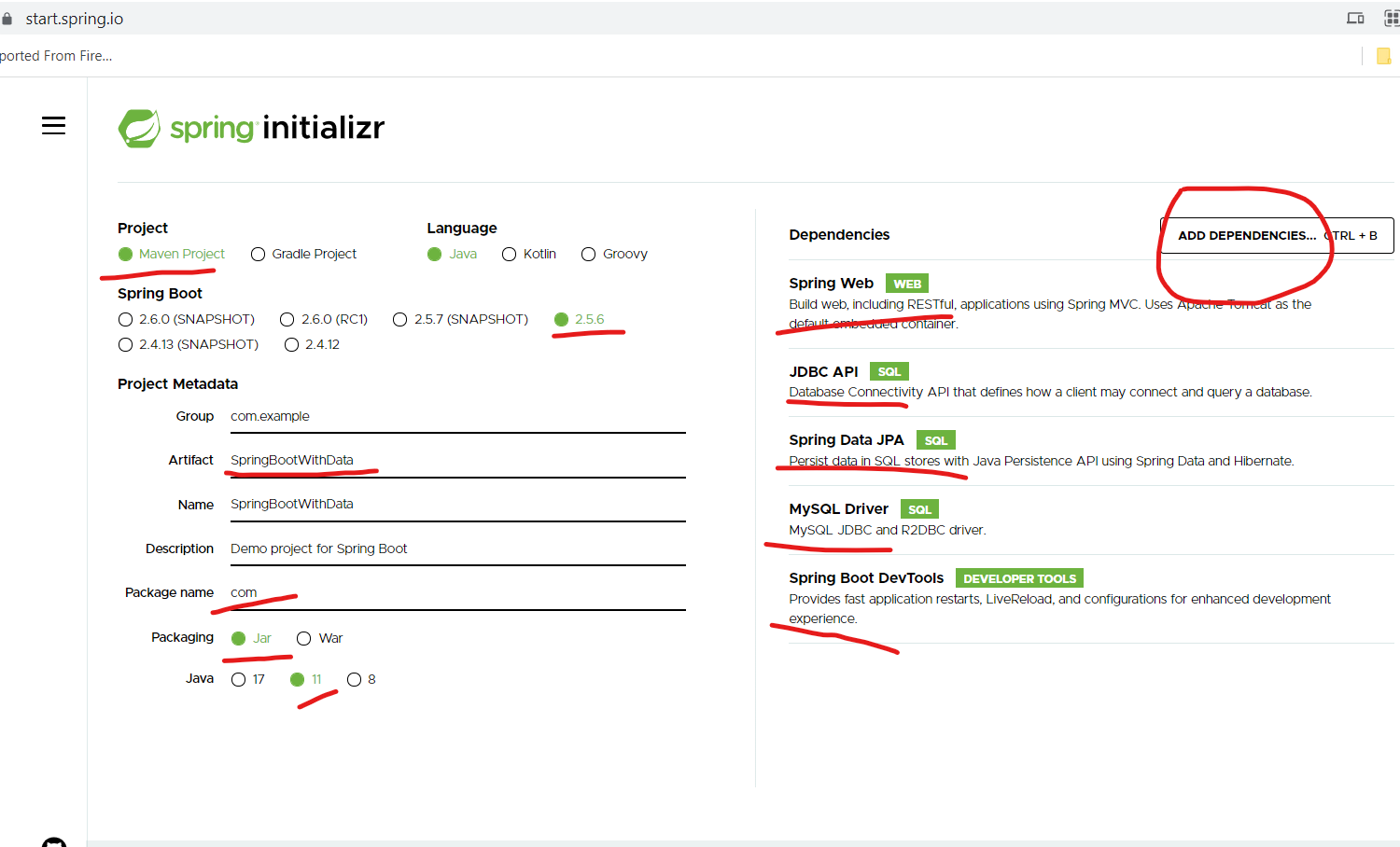
Spring Data :

Spring Data provide the abstraction for DAO layer.

In DAO layer we have to make one interface and it must be extends JpaRepository interface with generic Entity class 1st parameter and 2nd parameter Wrapper class name of column which contains primary key.

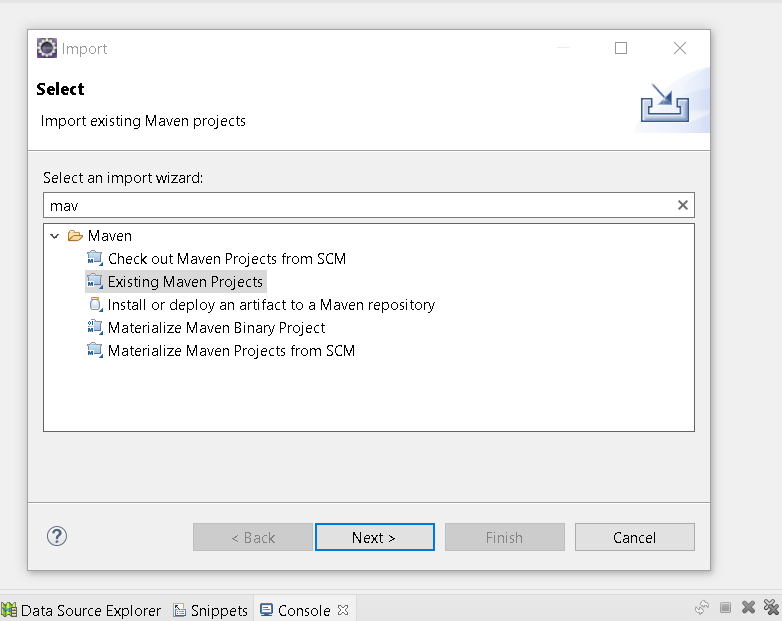
Spring framework provide their own website ie

Spring initializer which help to create the project very easily and we can add the required starter.



After selected all option click on generate button.

Then extra it and import in eclipse using maven project options.



AdminController

<http://localhost:9090/api/admin/signIn>

Admin signIn

Add Product

<http://localhost:9090/api/admin/addProduct>

Delete Product

View Product

Update Product

getProduct ById

UserController

Get ProductId

View Product

Sign Up

Sign In

Service layer

Dao layer

Using JPA or Using Spring Data