Phase 3 : Spring Framework

JEE : Servlet, JSP and EJB

Java Enterprise Edition : Servlet, JSP (Java Server Pages ) and EJB (Enterprise Java Bean)

View -🡪 HTML and JSP : Presentation Logic

Controller -🡪 Servlet

Model -🡪 Bean class or Entity class Using EJB We were improve model layer

Service class

Dao class (using JDBC or Hibernate)

Resource class (can be java class or xml file)

To improve the model layer we were using EJB. EJB is use to improve the model layer.

EJB was very complex. To develop the ejb program we require big server like jboss or web logic and complexity more to develop ejb application.

If we want to improve Model layer we have to make all model class is type of ejb. To run ejb program we require ejb container which is part of application server.

Spring Framework : spring framework is open source light weighted layer architecture framework.

Spring framework provided lot of pre-defined modules those module is use to improve every layer of application base upon our requirement.

Spring core

Spring context

Spring MVC

Spring DAO

Spring Rest

Spring security

Spring cloud

Etc

IOC : Inversion of Control : IOC is a concept or programming design pattern. According to IOC in place of creating any object or maintaining any resources allow to create by container. Pull from container whenever require. Use it and leave it. Container will maintain the life of the resources. Spring provide use features to create the object for normal class. That class is known as POJO class (Plain Old Java Object).

DI : Dependency Injection

Way to pull the object from container.

1. Constructor base di
2. Setter base di

We can achieve DI and IOC using XML configuration or annotation.

Singleton design pattern : if we want to create only one memory. Then we can use

Singleton design pattern.

Spring autowired : Autowired is a features of Spring framework which help us to enable you to inject the object (complex object ) dependency implicitly. By default di happen for primitive property implicitly not for complex object. If you want to achieve we have to take the help of auto wired features.

byType : when we use byType autowired in xml file it must contains only one bean tag for that type.

byName : if you want more than one bean tag then we have to use byname. In byname type id name and reference name must be match.

DI using Annotation

@Component we have to write on POJO class

@Autowried we have to write on complex property.

By default @Component annotation is not enable we have to enable using XML file or configuration class with few annotation.

If class name is one word then id is in lower case like for Employee class id is employee. If class name contains more than one word then id is Like EmployeeDetails then id is employeeDetails.

Query qry = session.createQuery(“select t.tname, t.tech, s.sname from Trainer t, Student s where t.tid = s.tsid”)

**28-06-2022**

Spring framework with datasource

Data source : it is use to provide the source of data for database connection using singleton design pattern like Connection pooling.

mysql –u root –p

Simplilearn

Use may\_bank

Select \* from product;

@Component ---🡪 JavaBean or POJO

@Service ---🡪 for Service class

@Repository --🡪 for Dao class

@Autowired -🡪 for complex property it may be user-defined or pre-defined

Product bean Product bean

DemoTest.java --------------🡪 Service layer -------------🡪Dao layer ( DataSource ) ---------------🡪 resource layer (beans.xml) file

Spring framework with DataSource ( we improve the Data Connectivity)

Spring DAO : Data Access Object : spring framework provided pre-defined API ie JdbcTemplate. It is pre-defined which help to do all jdbc operation in dao layer with simplicity.

JdbcTemplate wrap the Jdbc coding.

JdbcTemplate provided pre-defined interface ie RowMapper. This interface provide life cycle method which help to convert each record into object.

Spring MVC : Model View Controller : Spring MVC internally follow MVC architecture. It provided lot of annotation to improve controller and model layer. Spring MVC provide front controller class is DispatcherServlet. This class controller complete flow of the application. This controller we have to configure in web.xml file.



Handler mapping xml file configuration replace by @RequestMapping annotation. This annotation map our request path.

@Controller. This annotation is use to make the class behave like a controller. It is a like a servlet.

@Controller // it is like a servlet

public class MyController {

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

public ModelAndView sayHello() {

// coding

ModelAndView mav = new ModelAndView();

mav.setViewName(“display.jsp”); // like a forward

meturn mav;

}

}