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