Spring Framework or Adv Spring Framework

28 hours training

Spring boot, Spring reactive modules, spring web client, spring with hibernate.

Spring transaction management, spring security etc.

Spring framework

Core Java, JEE (Servlet and JSP) etc.

Basic programming, servlet and jsp

Oops

Exception handling

Multi threading

Collection framework

Jdbc

JEE : Java Enterprise Edition : which generally use to develop web application

Servlet Controller

JSP/html : Java Server pages View :

EJB : Enterprise Java Bean : Model : Bean class, Service class, Dao class, Resource class

MVC : Model View Controller

Framework : framework provide lot of api (application programming interface) which internally connected to each other to do specific task. If we develop any application using any framework 70 to 80% task taken care by framework. But framework is not a final product. It is template which help to develop the application very easily so we need to add 20 to 30% code to make final product. Framework internally follow standard. Implementation of design pattern is taken care by framework.

Struts : open source web framework provided by Apache. Internally follow MVC architecture. Provided lot of API to improve view layer, controller, and model layer but Struts mainly focus to improve controller layer.

Struts is known as Controller centric framework

JSF : java Server faces : provided by oracle. JSF is known as view centric framework

Angular or React JS Vs JSF

Hibernate : ORM framework. Limitation of JDBC resolve by Hibernate.

DAO : data access object layer.

Spring framework : spring is an open source layer architecture framework. It provided lot of layer to improve each module of the application. It also known as onion layer architecture framework.

EJB Vs Spring framework

Spring modules

1. Spring core
2. Spring context
   1. Spring mvc : spring MVC is known as model centric framework.
3. Spring jdbc or DAO
4. Spring orm
   1. Spring rest
5. Spring security
6. Spring micro service
7. Spring aop
8. Spring cloud
9. Spring security
10. Spring flux or reactive programming

Etc

Angular framework

Django framework

Express JS Framework

Spring core and context

IOC : Inversion of control : IOC is a concept. IOC is programming design pattern. According to IOC in place of creating any resource ie object creating, file handling, data base, security explicitly allow to create to container. If container create those resource it maintain properly. On demand pull it and use it and leave it.

Container : run time environment.

JRE

Web Container part of tomcat to run server and jsp

Ejb container to run ejb program part of application server

Docker container : responsible to run docker image

DI : Dependency Injection

The implementation of IOC is taken care by DI.

Pull the object or resource from container using

Type of DI

Constructor base DI

Setter base DI

To achieve the DI we need to configure using XML or annotation.

Constructor base and setter base DI using XML configuration

We use build tool as

1. Maven xml base
2. Gradle xml less

POJO : Plain Old Java Object :The class not to extends or implements any pre defined class

Web Container : part of tomcat server or any web server it will create object of servlet or jsp. If class is type of servlet or file must be type of .jsp then only web container create the object of those resource like servlet or jsp.

Struts : Stuts contains will create the object of that class if class is type of sturts etc.

But spring framework going to create the object of POJO class.

But we need to provide that class configuration details using xml or annotation.

Auto wired : spring container do the DI for primitive property implicitly. If class contains complex property then we need to use ref attribute part of property or constructor-arg tag to achieve di explicitly.

If we use auto wired features then we can achieve DI for complex property implicitly rather than explicitly using ref attribute.

When we use auto wired

1. byType : then spring container scan the xml file. If they found definition of that type then they inject it automatically.

In byType we need to provide only one bean definition for that type.

1. byName : in byName option we can write more than one bean definition of that type. But in byName id name and reference name part of class must be match.

@Component : This annotation we generally using on pojo class for generic purpose.

@Autowired : this annotation we need to use on complex property.

By default @Component annotation is not enable. We need to enable using

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