29 November 2021 21:33

In last session we discussed about @SpringBootApplication annotation & Component Scanning

=> @SpringBootApplication annotation is equal to below 3 annotations @SpringBootConfiguration

@EnableAutoConfiguration

@ComponentScan

What is Component Scanning?

-> Component Scanning is the process of identifying Spring Bean classes available in the application.

-> Component Scan will start from base package

-> After Base Package scanning got completed, it will scan sub packages of base package.

-> We can configure our own base package using @ComponentScan annotation in Spring Boot start class

How to represent Java class as Spring Bean ?

To represent java class as Spring Bean we will use below annotations

@Component @Service @Repository

@Controller @RestController

@Configuration @Bean

=> @Component is a stereo type annotation which is used to represent java class as Spring Bean.

=> By using @Service & @Repository also we can represent java class as Spring Bean. These annotations are also called as Stereotype annotation

=> If our java class contains business logic then we will use @Service annotation to represent that class as Spring Bean.

=> If our java class contains persistence logic then we will use @Repository annotation to represent that class as Spring Bean.

=> @Controller annotation is used to represent our java class as Spring Controller. In web applications we will use Controller to handle request & response.

=> @RestController annotation is used to represent our java class as Distributed Component. It is used to establish communication from one application to another application.

-> @Configuration annotation is used to represent our java class as Configuration class.

-> Configuration class is replacement for XML File

-> @Bean annotation is method level annotation. If we want to create object for any class with our own implementation then we will use @Bean annotation. The method which is annotated with @Bean will be called by IOC.

public class Engine {

public Engine() {

System.out.println("\*\* Engine :: Constructor \*\*");

}

}

@Configuration

public class AppConfig {

public AppConfig() {

System.out.println("\*\* AppConfig :: Constructor \*\*");

}

@Bean

public Engine getInstance() { Engine eng = new Engine(); return eng;

}

}

Note: @Component, @Service, @Repository, @Controller & @RestController annotations are class level annotations. @Bean annotation is method level annotation.