**Spring Boot Framework Annotations**

1. **@SpringBootApplication** – It is convenience annotation it combines other annotations like @EnableAutoConfiguration, @ComponentScan, and @Configuration.
2. **@EnableAutoConfiguration** – This automatically configure the spring boot application based on the jar dependency that we have added in our pom.xml file.

**Ex**: If the H2 database jar is present in the class path and we are not configured any beans related to the database manually, spring boot Autoconfiguration feature automatically configure it in our project.

1. **@ComponentScan** – This is just a simple annotation; it will just scan our bean so that it can be visible to the IOC container.

**Ex**: We are defining a different package structure want to scan specific package and class to IOC container then we can use this annotation.

1. **@Configuration** – We use this annotation for java-based configuration where we can define the bean definitions inside this class so that spring IOC can load it.

So, these three annotations spring boot developers combines to a single annotation that is the root annotation @SpringBootApplication.

**Stereotype annotations:**

Basically, these annotations are used to create spring bean automatically the application context if we annotated any of this annotation on top our spring bean, then spring will scan that bean and manage its life cycle begin from object creation to object destroy.

1. **@Component** – It is the main or base stereotype annotation we can say and other three are just derived from @Componenet annotation.
2. **@Service**
3. **@RestController/@Controller**
4. **@Repository**

We might have a question why these four annotations have derived from @Component annotation?

* We can use @Component annotation using of these four annotations, but the main purpose of other annotations will tell the role of that classes.
* @RestController or @Controller – Easily identify this is the web layer where we can expose our REST Apis.
* @Service – Easily identify this is the class where we write our business logic.
* @Repository – while seeing this annotation we are easily identify this is where write the database related operation.

**Spring Core relate annotations:**

1. **@Configuration and @Bean** – Usually these two annotations use when we want to use java-based configuration. It means we no need to handle the spring bean life cycle either using this annotations or xml.

So, when we write @Configuration it indicates that the class can be used by spring IOC container as a source of bean definition