Difference Between @Component, @Repository, @Service, and @Controller Annotations in Spring

Last Updated : 09 May, 2022

Spring Annotations are a form of metadata that provides data about a program. Annotations are used to provide supplemental information about a program. It does not have a direct effect on the operation of the code they annotate. It does not change the action of the compiled program. Here, we are going to discuss the difference between the 4 most important annotations in Spring, @Component, @Repository, @Service, and @Controller.

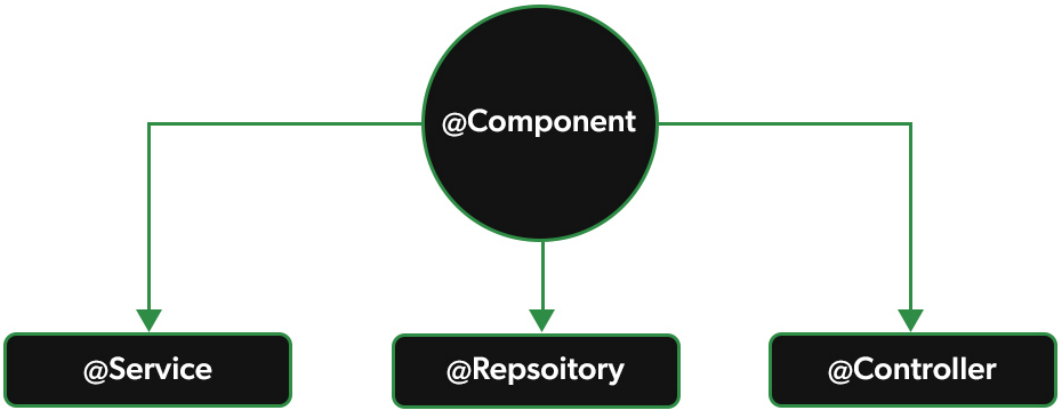
@Component Annotation

@Component is a class-level annotation. It is used to denote a class as a Component. We can use @Component across the application to mark the beans as Spring's managed components. A component is responsible for some operations. Spring framework provides three other specific annotations to be used when marking a class as a Component.

1. @Service
2. @Repository

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !



To read more about @Component Annotation refer to the article Spring @Component Annotation

A. @Service Annotation

In an application, the business logic resides within the service layer so we use the **@Service Annotation** to indicate that a class belongs to that layer. It is a specialization of **@Component Annotation**. One most important thing about the @Service Annotation is it can be applied only to classes. It is used to mark the class as a service provider. So overall @Service annotation is used with classes that provide some business functionalities. Spring context will autodetect these classes when annotation-based configuration and classpath scanning is used.

To read more about @Service Annotation refer to the article Spring @Service Annotation

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

@Repository Annotation is also a specialization of **@Component** annotation which is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects. Though it is a specialization of @Component annotation, so Spring Repository classes are autodetected by the spring framework through classpath scanning. This annotation is a general-purpose stereotype annotation which very close to the DAO pattern where DAO classes are responsible for providing CRUD operations on database tables.

> *To read more about **@Repository Annotation** refer to the article Spring @Repository Annotation*

C. @Controller Annotation

Spring @Controller annotation is also a specialization of **@Component** annotation. The @Controller annotation indicates that a particular class serves the role of a **controller**. Spring Controller annotation is typically used in combination with annotated handler methods based on the @RequestMapping annotation. It can be applied to classes only. It's used to mark a class as a web request handler. It's mostly used with Spring MVC applications. This annotation acts as a stereotype for the annotated class, indicating its role. The dispatcher scans such annotated classes for mapped methods and detects **@RequestMapping** annotations.

> *To read more about **@Controller Annotation** refer to the article Spring @Controller Annotation*

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

One of the interesting queries that arise in front of a developer is, can @Component, @Repository, @Service, and @Controller annotations be used interchangeably in Spring or do they provide any particular functionality? In other words, if we have a Service class and we change the annotation from @Service to @Component, will it still behave the same way?

So in order to answer the same, it is with respect to scan-auto-detection and dependency injection for BeanDefinition, all these annotations (@Component, @Repository, @Service, and @Controller) are the same. We could use one in place of another and can still get our way around.

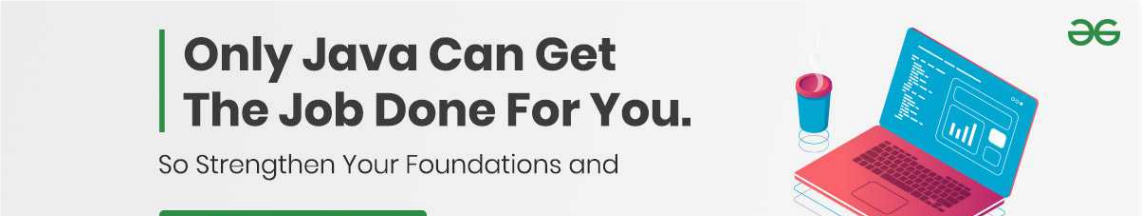
By now it is made clear that @Component is a general-purpose stereotype annotation indicating that the class is a spring component and @Repository, @Service, and @Controller annotations are special types of @Component annotation. Now let us finally conclude via plotting the ***difference table between all annotation types*** to grasp a better understanding that is as follows:

@Service Annotation	@Repository Annotation	@Controller Annotation
@Service annotation is used with classes that provide some business functionalities.	@Repository Annotation is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects.	@Controller annotation indicates that a particular class serves the role of a controller.

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

@Service Annotation	@Repository Annotation	@Controller Annotation
@Service Annotation is a specialization of @Component Annotation.	@Repository Annotation is also a specialization of @Component Annotation.	@Controller annotation is also a specialization of @Component annotation.
It is used to mark the class as a service provider.	It is used to mark the interface as DAO (Data Access Object) provider.	It's used to mark a class as a web request handler.
It is a stereotype for the service layer.	It is also a stereotype for the DAO layer.	It is also a stereotype for the presentation layer (spring-MVC).
Switch can be possible. But it is not recommended.	Switch can be possible. But it is not recommended.	We cannot switch this annotation with any other like @Service or @Repository.
It is a Stereotype Annotation.	It is also a Stereotype Annotation.	It is also a Stereotype Annotation.



We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

Like 0

Previous

How to Configure Dispatcher Servlet in Just Two Lines of Code in Spring?

Next

Java Program to Encrypt Password in Configuration Files

RECOMMENDED ARTICLES

Page : 1 2 3

01 Spring Boot - Difference Between @Service Annotation and @Repository Annotation
01, Feb 22

02 Difference Between @Controller and @Service Annotation in Spring
26, Apr 22

03 Create and Run Your First Spring MVC Controller in Eclipse/Spring Tool Suite
09, Mar 22

04 Difference Between @Controller and

05 Spring - Stereotype Annotations
01, Dec 21

06 Spring Framework Annotations
13, Dec 21

07 Spring Core Annotations
01, Feb 22

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

Article Contributed By :

AmiyaRanjanRout
@AmiyaRanjanRout

Vote for difficulty

Easy | Normal | Medium | Hard | Expert

Article Tags : Java-Spring, Difference Between, Java

Practice Tags : Java

Improve Article | Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

GeeksforGeeks
5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Company

About Us

Careers

In Media

Contact Us

Privacy Policy

Copyright Policy

Learn

Algorithms

Data Structures

SDE Cheat Sheet

Machine learning

CS Subjects

Video Tutorials

News

Top News

Technology

Work & Career

Business

Finance

Lifestyle

Languages

Python

Java

CPP

Golang

C#

SQL

Web Development

Web Tutorials

Django Tutorial

HTML

CSS

JavaScript

Contribute

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our Cookie Policy & Privacy Policy

Got It !