

[Open in app ↗](#)

♦ Member-only story

# What is difference between @Component and @Bean annotation in Spring?



Soma · [Follow](#)

Published in Javarevisited

7 min read · Apr 3

[Listen](#)[Share](#)[More](#)

## Spring Annotation



@Component  
vs  
@Bean

Hello folks, if you are preparing for Java Developer and Spring Developer interviews then you must prepare for questions like difference between X and Y like `@bean` vs `@component`. In the past, I have shared you about [difference between @Controller and @RestController annotation](#) and [@Controller vs @Service @Repository](#) and in this article, we will take a look at another set of popular Spring annotations.

In Spring, both `@Component` and `@Bean` annotations are used to register a bean in the Spring container, but they differ in the way they are used which you will find in this article.

By the way, if you are preparing for Java and programming interview then In my earlier articles, I have also shared [25 Advanced Java questions](#), [21 Software Design Pattern questions](#), [20 SQL queries from Interviews](#), [50 Microservices questions](#), [60 Tree Data Structure Questions](#), [15 System Design Questions](#), and [35 Core Java Questions](#) and [21 Lambda and Stream questions](#) which you can use for your Java interview preparation.

Now, coming back to the question, `@Component` is a class-level annotation that is used to mark a class as a Spring component, indicating that it should be automatically detected and registered as a bean in the Spring application context. The `@Component` annotation is a generic stereotype annotation that can be used to annotate any class.

On the other hand, `@Bean` is a method-level annotation that is used to explicitly declare a bean definition in a configuration class. It is typically used when a class does not have the `@Component` annotation, or when you want more fine-grained control over the configuration of the bean.

While `@Component` is used for automatic scanning and registering of beans, `@Bean` is used for explicit manual registration of beans. Another difference is that `@Component` is used for general classes while `@Bean` is used for creating a specific instance of a class and providing it to the Spring container.

In short, `@Component` is used to annotate a class and automatically detect and register it as a bean, whereas `@Bean` is used to declare a method that provides an instance of a bean to the Spring container.

Now, that you know the basic difference, its time for deep dive but, if you are not a Medium member yet then I highly recommend you to join Medium and read my other member only articles for your interview preparation. You can [join Medium here](#)

[Join Medium with my referral link - Soma](#)

Read this and every story from thousands of great writers on Medium). Read more from great writers on Medium like...

medium.com

## What is @Component in Spring?

in Spring, `@Component` is a generic annotation that can be used to indicate any Spring-managed component or bean. It serves as a base annotation for more specific annotations such as `@Service`, `@Repository`, and `@Controller`.

Here's an example of how `@Component` can be used to mark a class as a Spring-managed component:

```
@Component  
public class Order {  
    // class implementation  
}
```

By adding the `@Component` annotation to the `Order` class, Spring will automatically detect and manage this class as a bean within the application context. This allows it to be easily injected into other Spring-managed components using dependency injection.

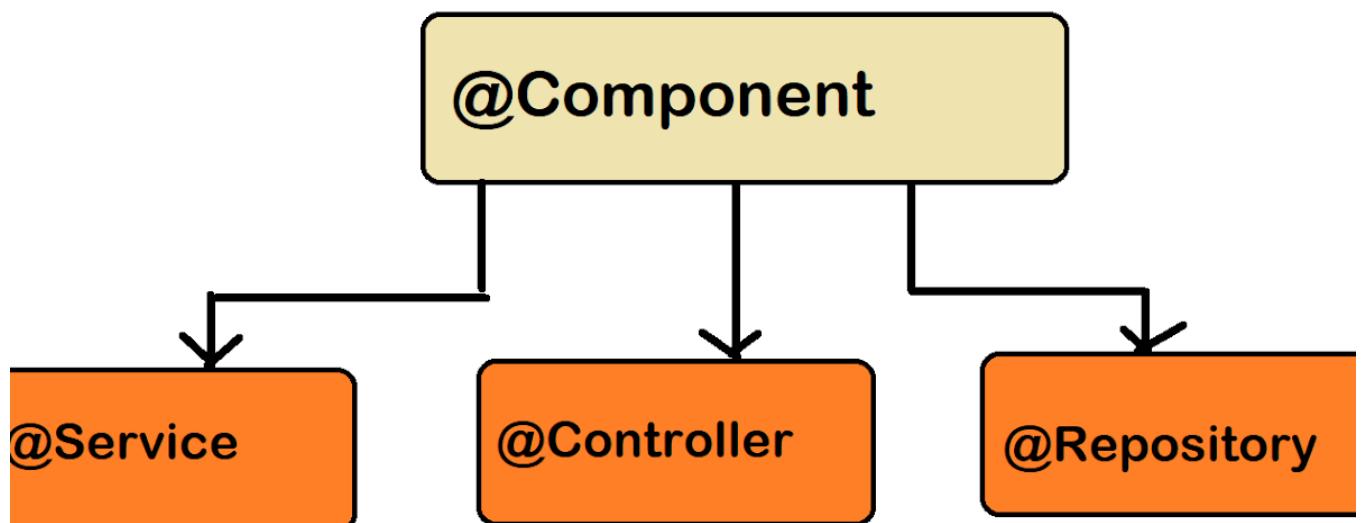
Here's an example of how a `@Component` can be used in conjunction with dependency injection:

```
@Component  
public class Invoice{  
    private final Order order;  
  
    @Autowired  
    public Invoice(Order order) {  
        this.order = order;  
    }  
}
```

```
// class implementation
}
```

In this example, `Invoice` is also a Spring-managed component that has a dependency on `Order`. By using the `@Autowired` annotation on the constructor, Spring will automatically inject an instance of `Order` into `Invoice` at runtime.

Overall, `@Component` is a powerful annotation that allows for easy management and dependency injection of Spring components.



### What is @Bean in a spring Framework?

In Spring, `@Bean` is an annotation that is used to declare a single bean. It is generally used to configure third-party beans or beans that are not defined in the application context.

Here's an example of using `@Bean` to declare a single bean:

```
@Configuration
public class AppConfig {
```

```

@Bean
public MyService myService() {
    return new MyServiceImpl();
}

```

In this example, `@Bean` is used to create a bean for the `MyService` interface. The method `myService()` returns a new instance of `MyServiceImpl`, which is then registered as a bean in the application context.

`@Bean` can also be used to specify bean dependencies. Here's an example:

```

@Configuration
public class AppConfig {

    @Bean
    public MyService myService() {
        return new MyServiceImpl();
    }

    @Bean
    public MyController myController(MyService myService) {
        return new MyControllerImpl(myService);
    }
}

```

In this example, `@Bean` is used to declare a bean for the `MyController` interface, which has a dependency on `MyService`. The `myController()` method takes a parameter of `MyService` and uses it to create a new instance of `MyControllerImpl`, which is then registered as a bean in the application context.

The `MyService` bean is automatically injected into the `MyController` bean because of its dependency declaration in the method signature.

## Difference between @Bean and @Component annotation in Spring Framework?

Here are the key differences between `@Bean` and `@Component` annotations in Spring:

## 1. Functionality

- `@Bean` is used to explicitly declare a bean in Spring's application context.
- `@Component` is a general-purpose annotation used to indicate a Spring-managed component.

## 2. Scope

- `@Bean` is used to define custom bean instantiation logic, and can be used to configure the scope of a bean.
- `@Component` doesn't define the scope of the bean, it uses the default scope of Singleton.

## 3. Usage:

- `@Bean` is typically used in configuration classes that are annotated with `@Configuration` or `@Import`.
- `@Component` is typically used to mark classes as Spring-managed components that are picked up automatically by component scanning.

## 4. Return type

- `@Bean` methods can return any object, whereas `@Component` is typically used to annotate classes that should be instantiated as beans.

## 5. Configuration

- `@Bean` is used to configure specific instances of a bean or to inject dependencies between beans.
- `@Component` is used to mark a class as a Spring-managed component, and is used to auto-detect and configure beans in the application context.

In summary, `@Bean` is more flexible and is used to create custom bean instantiation logic and configure specific instances of a bean, whereas `@Component` is used to mark classes as Spring-managed components and auto-detect and configure beans in the application context.

## Java and Spring Interview Preparation Material

Before any Java and Spring Developer interview, I always use to read the below resources

### Grokking the Java Interview

[Grokking the Java Interview: click here](#)

I have personally bought these books to speed up my preparation.

You can get your sample copy [here](#), check the content of it and go for it

Grokking the Java Interview [Free Sample Copy]: [click here](#)



*If you want to prepare for the Spring Boot interview you follow this consolidated ebook, it also contains microservice questions from spring boot interviews.*

### Grokking the Spring Boot Interview

You can get your copy here — [Grokking the Spring Boot Interview](#)



That's all about **difference between @Bean and @Component annotation in Spring Framework**. Even though, both `@Bean` and `@Component` are important annotations in Spring and used to manage beans in a Spring application context. The key difference between the two is that `@Component` is used to auto-detect and configure beans while `@Bean` is used to explicitly declare beans.

If you have a third-party class that you cannot modify and would like to declare it as a bean, you can use the `@Bean` annotation. On the other hand, if you have control over the class and want to make it a Spring bean, you can use the `@Component` annotation.

In general, the choice of which annotation to use depends on the situation and the level of control you need over the bean's creation and management. It is important to understand the difference between `@Bean` and `@Component` to make the best use of them in your Spring applications.

By understanding the differences between these annotations, you will be better prepared for your Java and Spring interview and have a deeper understanding of the Spring framework.

Additionally, you can also prepare Microservices Questions like [difference between API Gateway and Load Balancer](#), [SAGA Pattern](#), [how to manage transactions in](#)

Microservices, and difference between SAGA and CQRS Pattern, they are quite popular on interviews.

And, if you are not a Medium member then I highly recommend you to join Medium and read great stories from great authors from real field. You can **join Medium here**

### Join Medium with my referral link - Soma

Read this and every story from thousands of great writers on Medium). Read more from great writers on Medium like...

[medium.com](https://medium.com/@javarevisited)

Other Java and Microservices articles you may like

### Top 10 Microservices Problem Solving Questions for 5 to 10 years experienced Developers

From Service Discovery to Security: 10 Problem-Solving Questions to Test the Microservices Skills of Developers with 5...

[medium.com](https://medium.com/@javarevisited)

### 10 Things to Keep in Mind while Designing and Developing Microservices

Best practices and considerations for building effective microservices architecture

[medium.com](https://medium.com/@javarevisited)

### What is Event Sourcing Design Pattern in Microservices Architecture? How does it work?

An Overview of Event Sourcing Design Pattern and Its Implementation in Microservice Architecture.

[medium.com](https://medium.com/@javarevisited)

Spring Boot

Spring Framework

Java

Programming

Development


[Follow](#)


## Written by Soma

4.1K Followers · Editor for Javarevisited

Java and React developer, Join Medium (my favorite Java subscription) using my link [https://medium.com/@somasharma\\_81597/membership](https://medium.com/@somasharma_81597/membership)

### More from Soma and Javarevisited

| Messaging Model  | Traditional                                       | Publish/Subscribe                                | Traditional                            |
|------------------|---|--|--|
| Scalability      | Clustering/Network of Brokers                     | Partitioning                                     | Clustering/Network of Brokers          |
| Performance      | Moderate  | High   | High                                   |
| Data Persistence | On Disk (default), In-memory                      | On Disk  | On Disk (default), Database            |
| Integration      | Programming Languages, Databases, Web Servers     | Data Processing Systems, Databases, Data Sources | JMS Clients, Apache Camel, Apache CXF  |
| Suitable For     | Strict Ordering, Reliable Delivery, Moderate-High | Streaming Data, High Message Rates               | High Data Durability, High Performance |

 Soma in Javarevisited

## Difference between RabbitMQ, Apache Kafka, and ActiveMQ

Hello folks, if you are preparing for Java Developer interviews along with Spring Boot, and Microservices, you should also prepare...

· 8 min read · May 19

208

4

...



 Ajay Rathod in Javarevisited

## Comprehensive Spring-Boot Interview Questions and Answers for Senior Java Developers: Series-25

Greetings everyone,

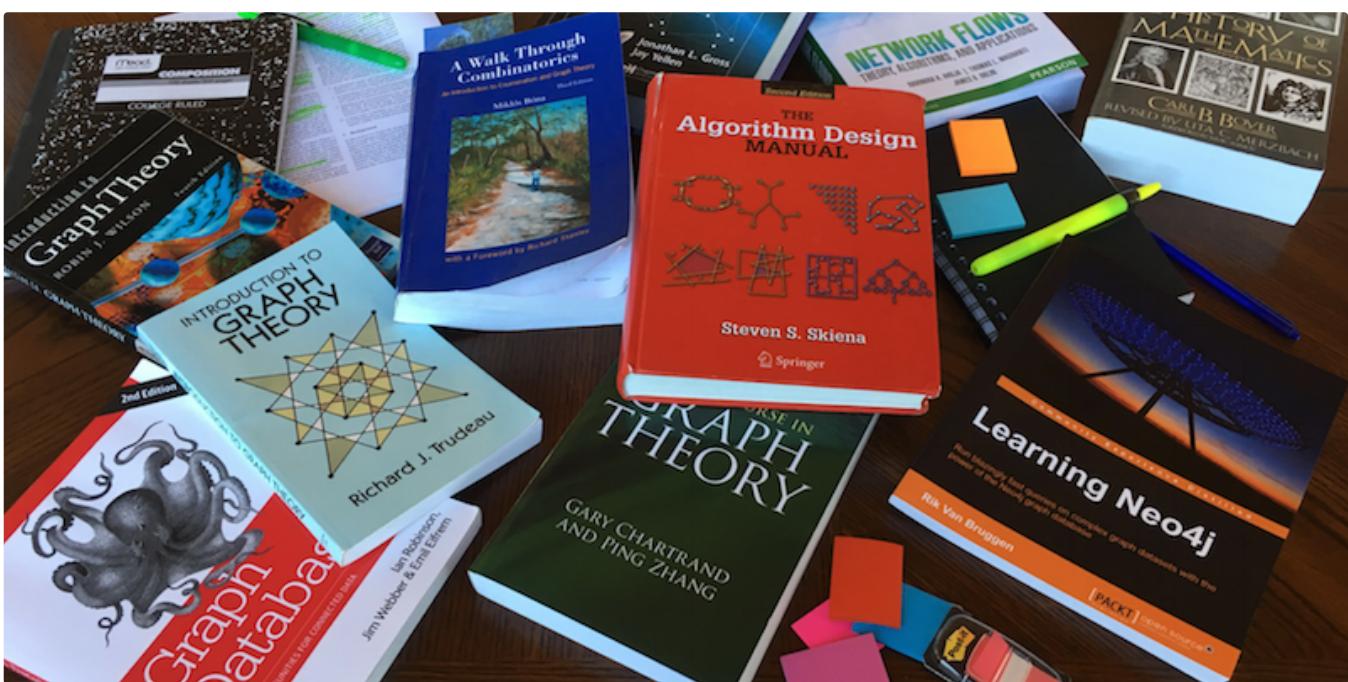
9 min read · 6 days ago

75

1



...





javinpaul in Javarevisited

## 10 Best Books for Data Structure and Algorithms for Beginners in Java, C/C++, and Python

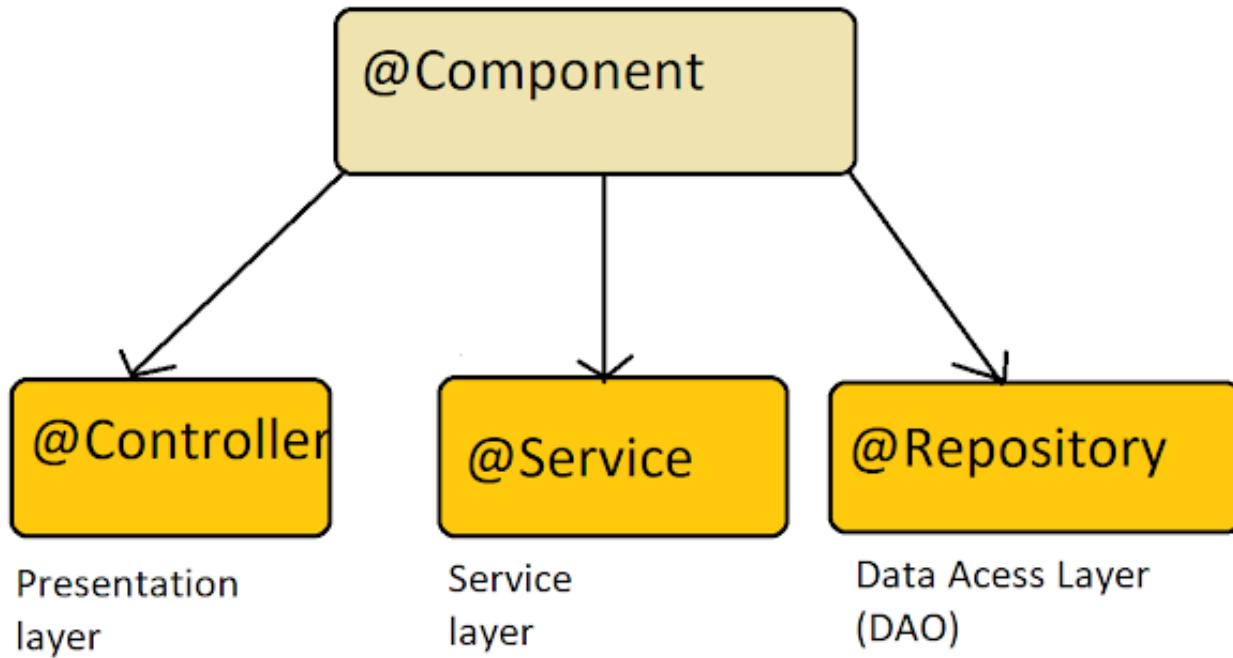
Algorithms are language agnostic, and any programmer worth their salt should be able to convert them to code in their programming language...

12 min read · Mar 13, 2020

1.2K 1



...



Soma in Javarevisited

## Difference between @Controller, @Service, and @Repository Annotations in Spring Framework?

While all three are stereotype annotation in Spring and can be used to represent bean where exactly they are used is the key for answering...

· 7 min read · Mar 23

238 1



...

[See all from Soma](#)

[See all from Javarevisited](#)

## Recommended from Medium



Hamza Nassour in Javarevisited

### What Happens Internally When You Start A Spring Boot Application(Part1)

ApplicationContext creation/registration , AutoConfiguration ...

4 min read · Feb 19

95



...



Rupert Waldron

## Create a non-blocking REST Api using Spring @Async and Polling

Get the great asynchronous, non-blocking experience you deserve with Spring's basic Rest Api, polling and @Aysnc annotation.

12 min read · Feb 26



22



...

### Lists



#### It's never too late or early to start something

13 stories · 69 saves



#### General Coding Knowledge

20 stories · 201 saves



#### Coding & Development

11 stories · 101 saves



#### Stories to Help You Grow as a Software Developer

19 stories · 269 saves



 Daniel Zielinski

## Senior Java Software Developer Interview Questions—part 1

1. Anemic Model vs Rich Model ?

◆ · 3 min read · Feb 18

 17 



...

The image shows the cover of a book titled "JAVA interview TOP MOST ASKED QUESTIONS 2023". The title is displayed in large, bold, sans-serif font. Below the title, the subtitle "TOP MOST ASKED QUESTIONS 2023" is written in a smaller, regular font. At the bottom of the cover, there are four company logos: Infosys (blue text), wipro (a colorful circular logo), TATA CONSULTANCY SERVICES (with a blue 'T' logo), and Deloitte (partially visible). The background of the book cover is white.

 Syed Habib Ullah

## Top Most Asked Java Interview Questions at Accenture, Infosys, Capgemini, Wipro, Cognizant...

As Java is one of the most popular programming languages used in software development today, it's no surprise that Java skill is highly...

9 min read · Apr 1

👏 30

💬 2



...

# Java Spring Boot Interview Questions



 Prabhash

## 35 very important interview questions

For Java, Spring Boot, Microservices Engineer Position

⭐ · 2 min read · Mar 5

👏 110

💬



...

 Daryl Goh

## Spring Boot: RequestEntity vs ResponseEntity | RequestBody vs ResponseBody

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can “just run”.

◆ · 5 min read · May 21

 40

...

---

See more recommendations