



# Spring Framework Annotations

Difficulty Level : Basic • Last Updated : 15 Dec, 2021

[Write & Earn](#) [DSA](#) [Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Data Science](#) [Top](#)

[Read](#)

[Discuss](#)

[Practice](#)

[Video](#)

[Courses](#)

[Spring framework](#) is one of the most popular [Java EE frameworks](#). It is an open-source lightweight framework that allows Java EE 7 developers to build simple, reliable, and scalable enterprise applications. This framework mainly focuses on providing various ways to help you manage your business objects. Now talking about Spring Annotation, 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. So in this article, we are going to discuss what are the main types of annotation that are available in the spring framework with some examples.

## Types of Spring Framework Annotations

Basically, there are 6 types of annotation available in the whole spring framework.

1. Spring Core Annotations
2. Spring Web Annotations
3. Spring Boot Annotations
4. Spring Scheduling Annotations
5. Spring Data Annotations
6. Spring Bean Annotations

### Type 1: Spring Core Annotations



# Start Your Coding Journey Now!

[Login](#)[Register](#)

Spring annotations present in the ***org.springframework.beans.factory.annotation*** and ***org.springframework.context.annotation*** packages are commonly known as Spring Core annotations. We can divide them into two categories:

- DI-Related Annotations
  - @Autowired
  - @Qualifier
  - @Primary
  - @Bean
  - @Lazy
  - @Required
  - @Value
  - @Scope
  - @Lookup, etc.
- Context Configuration Annotations
  - @Profile
  - @Import
  - @ImportResource
  - @PropertySource, etc.

## A DI (Dependency Injection) Related Annotations

### 1.1: @Autowired

@Autowired annotation is applied to the fields, setter methods, and constructors. It injects object dependency implicitly. We use @Autowired to mark the dependency that will be injected by the Spring container.

# Start Your Coding Journey Now!

## Java

```
class Student {  
    @Autowired  
    Address address;  
}
```

### 1.3: Constructor injection

---

## Java

```
class Student {  
    Address address;  
  
    @Autowired  
    Student(Address address) {  
        this.address = address;  
    }  
}
```

### 1.4: Setter injection

## Java

```
class Student {  
    Address address;
```



# Start Your Coding Journey Now!

```
}  
}
```

## B Context Configuration Annotations

@Profile: If you want Spring to use a @Component class or a @Bean method only when a specific profile is active then you can mark it with @Profile.

```
@Component  
@Profile("developer")  
public class Employee {}
```

## Type 2: Spring Web Annotations

Spring annotations present in the ***org.springframework.web.bind.annotation*** packages are commonly known as Spring Web annotations. Some of the annotations that are available in this category are:

- @RequestMapping
- @RequestBody
- @PathVariable
- @RequestParam
- Response Handling Annotations
  - @ResponseBody
  - @ExceptionHandler
  - @ResponseStatus
- @Controller
- @RestController
- @ModelAttribute
- @CrossOrigin

### Example: @Controller

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

# Start Your Coding Journey Now!

---

## Java

```
package com.example.demo.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

@Controller
public class DemoController {

    @RequestMapping("/hello")
    @ResponseBody
    public String helloGFG()
    {
        return "Hello GeeksForGeeks";
    }
}
```

### Type 3: Spring Boot Annotations

Spring annotations present in the ***org.springframework.boot.autoconfigure*** and ***org.springframework.boot.autoconfigure.condition*** packages are commonly known as Spring Boot annotations. Some of the annotations that are available in this category are:

- @SpringBootApplication
- @EnableAutoConfiguration
- Auto-Configuration Conditions
  - @ConditionalOnClass, and @ConditionalOnMissingClass
  - @ConditionalOnBean, and @ConditionalOnMissingBean
  - @ConditionalOnProperty
  - @ConditionalOnResource
  - @ConditionalOnWebApplication and @ConditionalOnNotWebApplication
  - @ConditionalExpression
  - @Conditional

 **Example:** @SpringBootApplication

# Start Your Coding Journey Now!

annotations with their default attributes.

---

## Java

```
@SpringBootApplication
```

```
// Class
```

```
public class DemoApplication {
```

```
    // Main driver method
```

```
    public static void main(String[] args)
```

```
    {
```

```
        SpringApplication.run(DemoApplication.class, args);
```

```
    }
```

```
}
```

## Type 4: Spring Scheduling Annotations

Spring annotations present in the ***org.springframework.scheduling.annotation*** packages are commonly known as Spring Scheduling annotations. Some of the annotations that are available in this category are:

- @EnableAsync
- @EnableScheduling
- @Async
- @Scheduled
- @Schedules

### Example: @EnableAsync

This annotation is used to enable asynchronous functionality in Spring.

```
@Configuration
```

```
@EnableAsync
```

```
class Config {}
```

## Type 5: Spring Data Annotations

Spring Data provides an abstraction over data storage technologies. Hence the business logic code can be much more independent of the underlying persistence

# Start Your Coding Journey Now!

- @Transactional
- @NoRepositoryBean
- @Param
- @Id
- @Transient
- @CreatedBy, @LastModifiedBy, @CreatedDate, @LastModifiedDate
- Spring Data JPA Annotations
  - @Query
  - @Procedure
  - @Lock
  - @Modifying
  - @EnableJpaRepositories
- Spring Data Mongo Annotations
  - @Document
  - @Field
  - @Query
  - @EnableMongoRepositories

## Example:

### A @Transactional

When there is a need to configure the transactional behavior of a method, we can do it with @Transactional annotation.

```
@Transactional
void payment() {}
```

**B @Id:** @Id marks a field in a model class as the primary key. Since it's implementation-independent, it makes a model class easy to use with multiple data store engines.

```
class Student {
```

```
    @Id
    Long id;
```



# Start Your Coding Journey Now!

```
// .....
```

```
}
```

## Type 6: Spring Bean Annotations

There're several ways to configure beans in a Spring container. You can declare them using XML configuration or you can declare beans using the `@Bean` annotation in a configuration class or you can mark the class with one of the annotations from the ***org.springframework.stereotype*** package and leave the rest to component scanning. Some of the annotations that are available in this category are:

- `@ComponentScan`
- `@Configuration`
- [Stereotype Annotations](#)
  - [@Component](#)
  - `@Service`
  - [@Repository](#)
  - [@Controller](#)

## Example: Stereotype Annotations

Spring Framework provides us with some special annotations. These annotations are used to create Spring beans automatically in the application context. `@Component` annotation is the main Stereotype Annotation. There are some Stereotype meta-annotations which is derived from `@Component` those are

1. `@Service`
2. `@Repository`
3. `@Controller`

**1: @Service:** We specify a class with `@Service` to indicate that they're holding the business logic. Besides being used in the service layer, there isn't any other special use for this annotation. The utility classes can be marked as Service classes.

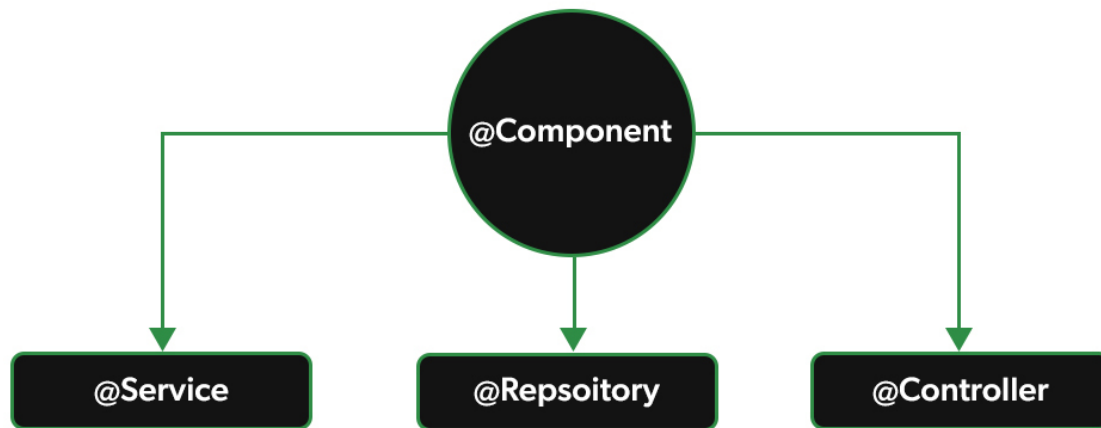
**2: @Repository:** We specify a class with `@Repository` to indicate that they're dealing with **CRUD operations**, usually, it's used with DAO (Data Access Object) or Repository implementations that deal with database tables.



## Start Your Coding Journey Now!

response. It is mostly used with REST Web Services.

*So the stereotype annotations in spring are **@Component**, **@Service**, **@Repository**, and **@Controller**.*



**MASTER  
JAVA BACKEND**

- ✓ Live Lectures
- ✓ 3 Projects
- ✓ Weekend Classes

**Enrol Now**

  
GeeksforGeeks

**Like** 12

[Previous](#)

[Next](#)



### Related Articles

# Start Your Coding Journey Now!

1. Spring Framework Annotations
2. Spring Core Annotations
3. Spring Boot - Annotations
4. Difference Between @Component, @Repository, @Service, and @Controller Annotations in Spring
5. Spring Boot - Spring JDBC vs Spring Data JDBC
6. Difference Between Spring DAO vs Spring ORM vs Spring JDBC
7. Aspect Oriented Programming and AOP in Spring Framework
8. Introduction to the Spring Data Framework
9. Spring - ORM Framework
10. Top 10 Most Common Spring Framework Mistakes

## Article Contributed By :



**AmiyaRanjanRout**  
@AmiyaRanjanRout

## Vote for difficulty

Current difficulty : [Basic](#)

Easy

Normal

Medium

Hard

Expert

Article Tags : [Java-Spring](#), [Java](#)

Practice Tags : [Java](#)

# Start Your Coding Journey Now!

[Improve Article](#)[Report Issue](#)**GeeksforGeeks**

A-143, 9th Floor, Sovereign Corporate Tower,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

## Company

[About Us](#)[Careers](#)[In Media](#)[Contact Us](#)[Privacy Policy](#)[Copyright Policy](#)[Advertise with us](#)

## News

[Top News](#)[Technology](#)[Work & Career](#)[Business](#)[Finance](#)[Lifestyle](#)[Knowledge](#)

## Learn

[DSA](#)[Algorithms](#)[Data Structures](#)[SDE Cheat Sheet](#)[Machine learning](#)[CS Subjects](#)[Video Tutorials](#)[Courses](#)

## Languages

[Python](#)[Java](#)[C++](#)[Golang](#)[C#](#)[SQL](#)[Kotlin](#)

## Web Development

[Web Tutorials](#)[Django Tutorial](#)

## Contribute

[Write an Article](#)[Improve an Article](#)

# Start Your Coding Journey Now!

Bootstrap  
ReactJS  
NodeJS

Internships  
Video Internship

@geeksforgeeks , Some rights reserved

