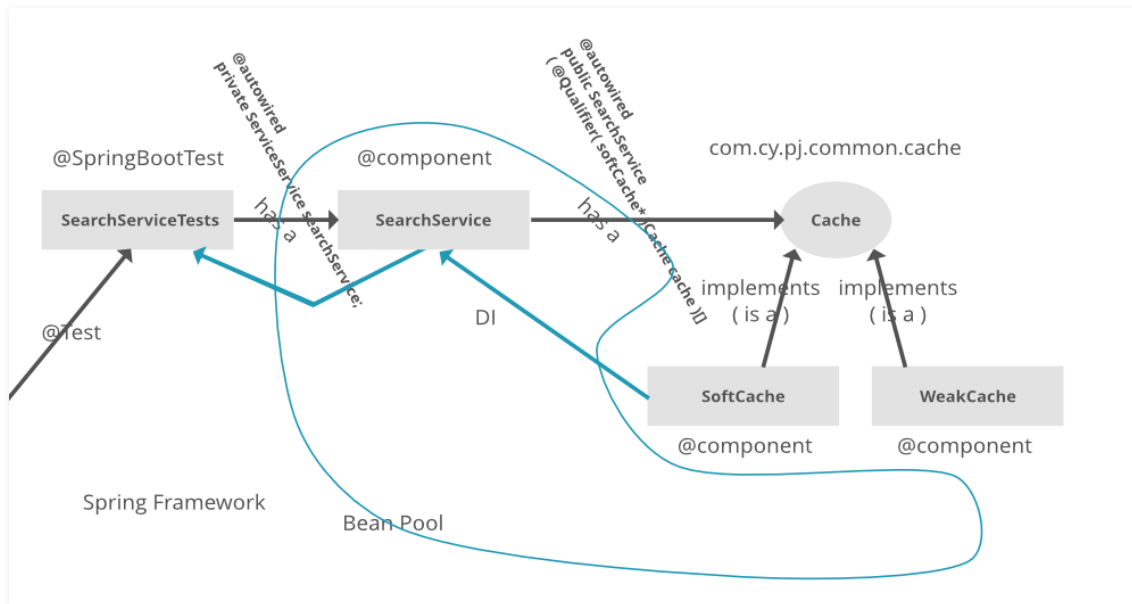


Spring @Autowired Annotation

Last Updated : 11 Mar, 2022

The @Autowired annotation marks a Constructor, Setter method, Properties and Config() method as to be autowired that is 'injecting beans'(Objects) at runtime by Spring Dependency Injection mechanism which is clearly depicted from the image below as shown:



Enabling @Autowired annotation

Spring beans can be declared either by Java configuration or XML configuration. By declaring beans, you provide metadata to the Spring Container to return the required dependency object at runtime. This is called Spring Bean Autowiring. In java based configuration, all the bean methods are defined in the class with **@configuration** notation. At runtime, Spring will provide bean definitions by reading those methods. Using **@Autowired**, the right dependency is assigned by the Spring Container.

Start Your Coding Journey Now!

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

```
}
```

In XML based configuration, if the beans are wired using **@Autowired annotation**, then **<context:annotation-config/>** has to be added to the XML file. Otherwise, you can include the **AutowiredAnnotationBeanPostProcessor** bean in the XML configuration file.

```
<bean class="org.springframework.beans.factory.annotation.AutowiredAr
```

The **@SpringBootApplication** which is a combination of **@Configuration**, **@EnableAutoConfiguration** and **@ComponentScan** scans all the components or services and other configuration files included in the base and child packages. This will register them in Spring Context and inject the beans at runtime using **@Autowired**.

```
@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

Using @Autowired

XML Configuration file

If XML configuration is used to wire the beans, then the configuration file looks like this,

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans/beans.xsd
http://www.springframework.org/schema/beans/spring-beans.xsc
```

Start Your Coding Journey Now!

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

```
<context:annotation-config/>
<bean id="customer" class="com.gfg.demo.domain.Customer">
    <property name="type" value="1" />
</bean>

<bean id="person" class="com.gfg.demo.domain.Person">
    <property name="name" value="ganesh" />
    <property name="age" value="21" />
</bean>
</beans>
```

Java Configuration class

If Java Configuration is used to wire the beans, then the configuration class looks like this,

```
package com.gfg.demo.config;

import com.gfg.demo.domain.Customer;
import com.gfg.demo.domain.Person;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class AppConfig {
    @Bean
    public Person person(){
        Person person = new Person();
        person.setName("ganesh");
        person.setAge("21");
        return person;
    }
}
```

Start Your Coding Journey Now!

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

The beans can be wired via constructor or properties or setter method. For example, there are two POJO classes Customer and Person. The Customer class has a dependency on the Person.

Customer.java

```
@Component
public class Customer {

    private int type;
    private Person person;

    // Constructors
    // getters and setter
}
```

Person.java

```
public class Person {
    private String name;
    private String age;

    // Constructors
    // getters and setters
}
```

Constructor based Autowiring

@Autowired annotation is optional for constructor based injection. Here, the person object from the container is passed to the constructor while creating the Customer object.

```
@Component
public class Customer {
```



Start Your Coding Journey Now!

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

```
,  
  
@Autowired  
public Customer(Person person) {  
    this.person = person;  
}  
}
```



[Data Structures](#) [Algorithms](#) [Interview Preparation](#) [Topic-wise Practice](#) [C++](#) [Java](#) [Python](#)

@Autowired annotation

```
@Component  
public class Customer {  
  
    private int type;  
  
    @Autowired  
    private Person person;  
}
```

Setter based Autowiring

The setter method will be called with the Person object at runtime by the container.

```
@Autowired  
public void setPerson(Person person) {  
    this.person = person;  
}
```

Optional Dependencies

If the bean of type **Person** is not defined then Spring will throw

NoSuchBeanDefinitionException. It prevents the Spring Container **from** launching

Start Your Coding Journey Now!

Login

Register

To fix this, we can set the **required** attribute of @Autowired as **false**,

```
@Autowired(required = false)
private Person person;
```

**Only Java Can Get
The Job Done For You.**

So Strengthen Your Foundations and

Start Learning



Like 10

Previous

Next

RECOMMENDED ARTICLES

Page : 1 2 3



01

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

05

**Spring @Repository Annotation
with Example**
01, Dec 21



Start Your Coding Journey Now!

Login

Register

03 Spring - @PostConstruct and
@PreDestroy Annotation with
Example
07, Mar 22

07 Spring Data JPA - @Table
Annotation
24, Dec 21

04 Spring @Component Annotation
with Example
28, Nov 21

08 Spring Data JPA - @Id Annotation
24, Dec 21

Article Contributed By :



Ganeshchowdharysadanala
@Ganeshchowdharysadanala

Vote for difficulty

Easy

Normal

Medium

Hard

Expert

Improved By : iamsamar12

Article Tags : Java-Spring, Java

Practice Tags : Java

Improve Article

Report Issue



Start Your Coding Journey Now!

Login

Register



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

News

Top News
Technology
Work & Career
Business
Finance
Lifestyle

Learn

Algorithms
Data Structures
SDE Cheat Sheet
Machine learning
CS Subjects
Video Tutorials

Languages

Python
Java
CPP
Golang
C#
SQL

Web Development

Web Tutorials

Contribute

Write an Article

Start Your Coding Journey Now!

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

Bootstrap

Video Internship

@geeksforgeeks , Some rights reserved

Do Not Sell My Personal Information

