

@Component is an annotation that allows Spring to automatically detect our custom beans.

In other words, without having to write any explicit code, Spring will:

- Scan our application for classes annotated with *@Component*
- Instantiate them and inject any specified dependencies into them
- Inject them wherever needed

Spring has provided a few specialized stereotype annotations: *@Controller*, *@Service* and *@Repository*. They all provide the same function as *@Component*.

@Component exclusively for our bean auto-detection needs.

```
@Controller public class ControllerExample { }
@Service public class ServiceExample { }
@Repository public class RepositoryExample { }
@Component public class ComponentExample { }
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.RUNTIME)
@Component public @interface CustomComponent { }
@CustomComponent public class CustomComponentExample { }

@Component("fooFormatter")
public class FooFormatter {
    public String format() {
        return "foo";
    }
}
```

```
@Component  
  
public class FooService {  
  
    @Autowired  
    private FooFormatter fooFormatter;  
}
```

Spring Component Example

Let's create a very simple Spring maven application to showcase the use of Spring Component annotation and how Spring autodetects it with annotation-based configuration and classpath scanning.

Create a maven project and add following spring core dependency.

```
<dependency>  
    <groupId>org.springframework</groupId>  
    <artifactId>spring-context</artifactId>  
    <version>5.0.6.RELEASE</version>  
</dependency>
```

That's all we need to get the spring framework core features.

Let's create a simple component class and mark it with `@Component` annotation.

```
package com.journaldev.spring;

import org.springframework.stereotype.Component;

@Component
public class MathComponent {

    public int add(int x, int y) {

        return x + y;

    }

}
```

Now we can create an annotation based spring context and get the MathComponent bean from it.

```
package com.journaldev.spring;

import
org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class SpringMainClass {

    public static void main(String[] args) {

        AnnotationConfigApplicationContext context = new
AnnotationConfigApplicationContext();

        context.scan("com.journaldev.spring");

        context.refresh();

        MathComponent ms = context.getBean(MathComponent.class);

        int result = ms.add(1, 2);

        System.out.println("Addition of 1 and 2 = " + result);

        context.close();

    }

}
```