



Section 3 Part -3 How does the @ComponentScan works?

🌱 This annotation tells the spring that where to look for the components such as @Beans, @Services, @Repository and @Component.

🌱 We will see what are all the listed annotations in upcoming sections. For now understand them as classes and objects of different class.

🌱 Previously we have seen about the @Configuration annotation where i have said that it is used for manual configuration and the beans are the objects of the manually defined configuration class. Well this is correct. But ultimately component annotation takes care of all the components and inject them wherever it is required.

🌱 It ensures that spring can find and manage all the required components without the manual intervention.

What does this @ComponentScan annotation actually does?

1 Scans the specified packages and sub packages.

2 automatically detects the classes that are annotated with @Component, @Service, @Repository and @Controller.

There can be cases when the spring boot is unable to scan all the packages and the classes/components in them, So we have to explicitly tell spring boot to scan this specific classes during the runtime of application.

We say it in the main method's class by using the annotation of **@componentScan (←-giving the package name- →).**

```
@SpringBootApplication
@ComponentScan(basePackages = "com.example.services")
public class MySpringBootApplication {
    public static void main(String[] args) {
        SpringApplication.run(MySpringBootApplication.class, args);
    }
}
```

The class that we want to scan is under the package that we given in the brackets.

```
package com.example.services;

@Service
public class MyService { }
```