Spring Dependency Injection How

19 Temmuz 2018 Perşembe 13:24

If a class has depency via implementation, this class isn't testable and
Time by time class will be more complex and it will not be easy
To change. If we want to fix a problem, because of lack of test another
part will be break.
We have to program to interface approach.

With dependency injection
-Maintainable
-Reusable
-Testable

Code we will have.

How to Use

1- Start point is defining App.Config or application.xml
If we defined AppConfig we have to annoted it with @Configuration

An object inside Spring container known as Beans.

2- In Java based approach all Beans defined in App.Config. Method name is by default is defined as key to access to Spring beans. If we call methods several time it wont affect object creation. Because by default all Spring Beans are singleton. We can also inject dependency via giving method parameter. Spring 5 container will inject them automatic.

Good example for Spring BeanFactory custom impl
http://www.captaindebug.com/2011/06/implementing-springs-factorybean.html

Java based annotation approach breaks POJO model but easy to configurable. In this approach, methodName which is to create a bean instance name is key value in container.

Constructor injection vs setter injection
Constructor injection is good for immutability. We have to inject mandotory dependency via constructor. But it has circular problem.
It is more compact and clear to understand.
Setter Injection if good for configurable dependency changing and solves cycle problem. It is also good for creating heavy object when needed. But it violates immutablity and not easy to read.

StereoType
used for spesific behaviour at different layer of application

Autowire
It reduces verbosity in your code.
With this annotation container will wire dependecy auto
It can be via constructor, setter, property.
In case of dependecy not found, we will have an error.
In case we need disable this nature for some variable we can
Disable it via required=false annotation parameter

**In case of for same interface has many impl,
You can use @Qualifier("") at where you inject dependency