Dependency Injection in Spring

Dependency Injection (DI) is a design pattern that removes the dependency from the programming code so that it can be easy to manage and test the application. Dependency Injection makes our programming code loosely coupled. To understand the DI better, Let's understand the Dependency Lookup (DL) first:

Dependency Lookup

The Dependency Lookup is an approach where we get the resource after demand. There can be various ways to get the resource for example:

```
A obj = new AImpl();
```

In such way, we get the resource(instance of A class) directly by new keyword. Another way is factory method:

```
A obj = A.getA();
```

This way, we get the resource (instance of A class) by calling the static factory method getA().

Alternatively, we can get the resource by JNDI (Java Naming Directory Interface) as:

```
Context ctx = new InitialContext();
Context environmentCtx = (Context) ctx.lookup("java:comp/env");
A obj = (A)environmentCtx.lookup("A");
```

There can be various ways to get the resource to obtain the resource. Let's see the problem in this approach.

Problems of Dependency Lookup

There are mainly two problems of dependency lookup.

- **tight coupling** The dependency lookup approach makes the code tightly coupled. If resource is changed, we need to perform a lot of modification in the code.
- **Not easy for testing** This approach creates a lot of problems while testing the application especially in black box testing.

Dependency Injection

The Dependency Injection is a design pattern that removes the dependency of the programs. In such case we provide the information from the external source such as XML file. It makes our code loosely coupled and easier for testing. In such case we write the code as:

```
class Employee{
Address address;

Employee(Address address){
this.address=address;
}
public void setAddress(Address address){
this.address=address;
}

}
```

In such case, instance of Address class is provided by external souce such as XML file either by constructor or setter method.

Two ways to perform Dependency Injection in Spring framework

Spring framework provides two ways to inject dependency

- By Constructor
- By Setter method

Upcoming topics in Spring Dependency Injection

Dependency Injection by constructor

Let's see how we can inject dependency by constructor.

Dependency Injection by setter method

Let's see how we can inject dependency by setter method.

<<prev next>>