In Last session we discussed about Autowiring

01 December 2021   21:38

=> Autowiring is used to inform IOC to perform Dependency Injection

=> To perform Autowiring we will use @Autowired annotation

=> @Autowired annotation we can use at below 3 places
  i)   Setter method level
  ii) Constructor level
  iii) Field level

===

Can we access private variables from outside of the class in java ?

Yes, using Reflection api.

===

```
public class User {

    private Integer age;

    public Integer getAge() {
        return age;
    }
}
```

===

```
public class Demo {

    public static void main(String[] args) throws Exception {

        Class<?> clz = Class.forName("in.ashokit.User");

        Field field = clz.getDeclaredField("age");

        field.setAccessible(true);

        Object object = clz.newInstance();

        field.set(object, 25);

        User user = (User) object;

        System.out.println(user.getAge());
    }
}
```

===

-> If we use @Autowired annotation at setter method level then ioc will perform Dependency Injection by calling setter method

-> If we use @Autowired annotation at constructor then IOC will perform Dependency Injection by calling Constructor

-> If we use @Autowired annotation at field level then IOC will perform Dependency Injection through Reflection API.

=> Field Injections is very easy to use and less lines of code but it is not recommended because it is not respecting OOPS principles.

=> If we do setter injection & constructor injection for same variable then setter injection will override constructor injection hence setter injection value be the final value.

===

Banner in Spring Boot

===

-> When we run Spring Boot application we can spring logo on console that is called as banner

-> We have 3 modes of banner in Spring Boot , they are

     i) Console (It is default mode)
    ii) Log (It will print banner in log file)
   iii) Off (It will stop banner printing)

-> We can customize banner text by creating banner.txt file in src/main/resources folder

===