Spring - Inversion Of Control (IOC)

IOC- Container is a box where object initialization is done by spring. IOC container will create the wired or coupled object and use it throughout in spring life cycle. **In whole, it will manage the object from creation to its destruction.**

There are two types of containers.

1. BeanFactory interface (Subset)
2. ApplicationContext interface (Superset)

BeanFactory –

- It is used for achieving dependency injection (DI) in spring IOC.
- It is has demanded/Lazy initialization of the object.
- It is used for the smaller application, e.g., .jar file.

Below code is for how BeanFactory gives bean or object to us.

```
XmlBeanFactory factory = new XmlBeanFactory (new
ClassPathResource("beans.xml"));

TestClass obj = (TestClass) factory.getBean("testclass ");
```

ApplicationContext –

- It is the superset of BeanFactory, means it is implementing BeanFactory interface.
- It has its feature plus all features which BeanFactory gives.
- It has eager/early initialization of the object.
- It is also used for achieving dependency injection (DI) in spring IOC.
- It is used in large enterprise applications.

Below code is for how ApplicationContext gives bean or object to us.

```
ApplicationContext context=new
ClassPathXmlApplicationContext("beans.xml");

TestClass obj = (TestClass) context.getBean("testclass ");
```

Please add on comments and like this lesson, if it gave you some revision or refreshment on Spring - Inversion of control (IOC) concepts.