**Important note: Before going further you are requested to go through theory of spring bean lifecycle methods mentioned in JavaLive Notes on Spring Core thoroughly so that you can understand the concept very well. (\*\*This activity i.e. going through theory part of respective subject is applicable for each of our project.)**

**\*\*Also go through initial comment section of file 'com.javaLive.SpringCoreBeanLifeCycleDemo.Main.java' for more notes.**

**What this project does??**

This project explains how BeanPostProcessor interface plays role in spring bean life cycle.

**Note that this is xml based spring project having configuration file viz. ' spring-config.xml' in resources folder.**

**Steps to create project:-**

1. Create quick start maven project. Refer file 'CreateQuickStartMavenProjectInEclipse.docx' in 'SprinDIWithAutowiring' project for more details.

2. Add require dependencies for spring as shown in pom.xml file.

3. Create required packages and add the files.

**Functioning of the project**

This project is mainly divided in four parts:

1. com.javaLive.beans package contains a files Book.java and Publisher.java which declared as bean in spring-config.xml file.

\*\*Note that Book.java implements interfaces InitializingBean, DisposableBean, BeanFactoryAware, BeanNameAware, BeanClassLoaderAware. So that we can study sequence of the execution of these interfaces along with BeanPostProcessor

**2. ' com.javaLive.SpringCoreBeanLifeCycleDemo. MyBeanPostProcessor.java' is the most important class of this project. This class implements BeanPostProcessor interface.**

**Note that functionality in BeanPostProcessor is applicable for each bean defined in spring-config.xml.**

3. spring-config.xml file in resources folder contains bean definition for above class.

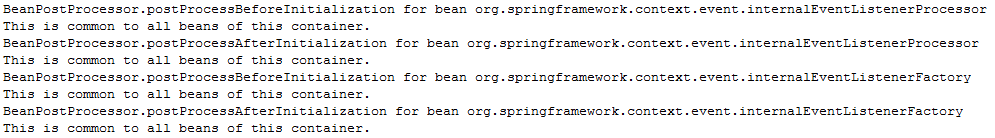
Note here it also provide registration for class that implements BeanPostProcessor interface viz. MyBeanPostProcessor.java which activates functioning of BeanPostProcessor interface for each bean defined in spring-config.xml file.

\*\*Another attribute of the bean definition Book and Publisher to pay attention is <lazy-init>. Please go through the comment section in spring-config.xml for more details on <lazy-init>.

4. com.javaLive.main package contains MainClass.java class which get the bean from spring application context and displays its details. When we run this file, we will come to know how BeanPostProcessor functions in three different scenarios as follows:

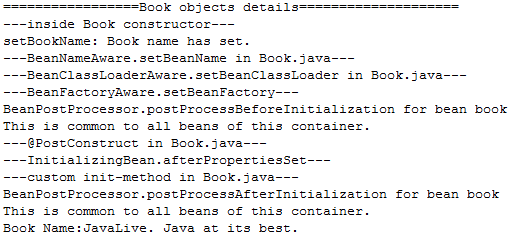
i) With system beans viz. org.springframework.context.event.internalEventListenerProcessor and org.springframework.context.event.internalEventListenerFactory.

Following is the screen shot of the output for above beans.



ii) With bean (viz. Book.java in this case) implementing InitializingBean, DisposableBean, BeanFactoryAware, BeanNameAware, BeanClassLoaderAware

Following is the screen shot of the output for above beans.





iii) With plain bean i.e. bean without any other interface implementation of bean life cycle. (viz. Publisher.java in this case.)

Following is the screen shot of the output for above beans.

