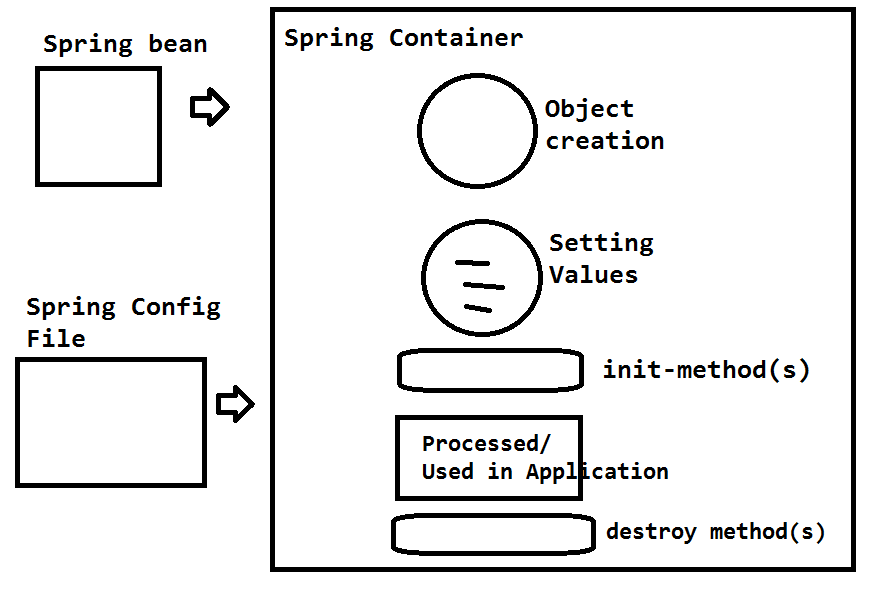
Spring Bean Life Cycle Methods:

Spring Bean is having two life cycle methods, which are called automatically by spring container.

1. init-method
2. destroy method

Note: init-method will be executed after creating bean (After CI and SI done). And Destroy method will be called once before the Container shutdown or bean is removed from container.



These are optional methods, an can be configured in 3 ways.

1. Using Interfaces
2. XML Configuration
3. Annotation Based
4. Using Interfaces: In Spring, to configure the init method InitializingBean which is having an abstract method afterPropertiesSet():void and DisposableBean having an abstract method destroy. We need to override these two methods in Bean class.

Ex:

**package** com.app.module;

**import** org.springframework.beans.factory.DisposableBean;

**import** org.springframework.beans.factory.InitializingBean;

**public** **class** Employee **implements** InitializingBean,DisposableBean{

**private** **int** empId;

**public** Employee() {

**super**();

System.*out*.println("In default constructor");

}

**public** **int** getEmpId() {

**return** empId;

}

**public** **void** setEmpId(**int** empId) {

**this**.empId = empId;

System.*out*.println("In empId setter method");

}

@Override

**public** String toString() {

**return** "Employee [empId=" + empId + "]";

}

@Override

**public** **void** afterPropertiesSet() **throws** Exception {

System.*out*.println("Init -method");

}

@Override

**public** **void** destroy() **throws** Exception {

System.*out*.println("Destory Method");

}

}

XML Config Code:-

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:c=*"http://www.springframework.org/schema/c"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean class=*"com.app.module.Employee"* name=*"empObj"*>

<property name=*"empId"* value=*"200"*/>

</bean>

</beans>

Test.java

**public** **class** Test{

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

AbstractApplicationContext context=**new** ClassPathXmlApplicationContext("config.xml");

Employee emp=(Employee)context.getBean("empObj");

System.*out*.println(emp);

context.registerShutdownHook();

}

}

1. XML Configuration:- By using <bean> tag specify two attributes, init-method and destroy-method and provide method names for these.

Ex:

**package** com.app.module;

**public** **class** Employee {

**private** **int** empId;

**public** Employee() {

**super**();

System.*out*.println("In default constructor");

}

**public** **int** getEmpId() {

**return** empId;

}

**public** **void** setEmpId(**int** empId) {

**this**.empId = empId;

System.*out*.println("In empId setter method");

}

@Override

**public** String toString() {

**return** "Employee [empId=" + empId + "]";

}

**public** **void** abcd(){

System.*out*.println("In init-method");

}

**public** **void** mnop(){

System.*out*.println("In destroy-method");

}

}

XMl Code:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:c=*"http://www.springframework.org/schema/c"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean class=*"com.app.module.Employee"* name=*"empObj"* init-method=*"abcd"* destroy-method=*"mnop"*>

<property name=*"empId"* value=*"200"*/>

</bean>

</beans>

\*\*\*Test Class Remains Same

1. Annotation Based:- Here two Annotations are used to they are: @PostConstruct (init) and @PreDestroy(destroy)

By Default all Annotations are in spring is disabled mode. We have to enable the annotation before using them, either specific or all annotations.

To Enable these two annotations configure bean class:

***org.springframework.context.annotation.CommonAnnotationBeanPostProcessor***

Ex:

**package** com.app.module;

**import** javax.annotation.PostConstruct;

**import** javax.annotation.PreDestroy;

**public** **class** Employee {

**private** **int** empId;

**public** Employee() {

**super**();

System.*out*.println("In default constructor");

}

**public** **int** getEmpId() {

**return** empId;

}

**public** **void** setEmpId(**int** empId) {

**this**.empId = empId;

System.*out*.println("In empId setter method");

}

@Override

**public** String toString() {

**return** "Employee [empId=" + empId + "]";

}

@PostConstruct

**public** **void** abcd(){

System.*out*.println("In init-method");

}

@PreDestroy

**public** **void** mnop(){

System.*out*.println("In destroy-method");

}

}

XML Code:

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:c=*"http://www.springframework.org/schema/c"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd"*>

<bean class=*"org.springframework.context.annotation.CommonAnnotationBeanPostProcessor"*/>

<bean class=*"com.app.module.Employee"* name=*"empObj"*>

<property name=*"empId"* value=*"200"*/>

</bean>

</beans>

Note: If we configure 3 ways , then container creates Object and calls 3 init and destroy methods in below order

1. Annotated Methods
2. Interface Methods
3. XML Configured Methods