

Spring init-method and destroy-method example

By mkyong (<http://www.mkyong.com/author/mkyong/>) | March 28, 2010 | Updated : June 13, 2011

In Spring, you can use **init-method** and **destroy-method** as attribute in bean configuration file for bean to perform certain actions upon initialization and destruction. Alternative to InitializingBean and DisposableBean interface (<http://www.mkyong.com/spring/spring-initializingbean-and-disposablebean-example/>).

Example

Here's an example to show you how to use **init-method** and **destroy-method**.

```
package com.mkyong.customer.services;

public class CustomerService
{
    String message;

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public void initIt() throws Exception {
        System.out.println("Init method after properties are set : " + message);
    }

    public void cleanUp() throws Exception {
        System.out.println("Spring Container is destroy! Customer clean up");
    }
}
```

File : Spring-Customer.xml, define **init-method** and **destroy-method** attribute in your bean.

```
<beans xmlns="http://www.springframework.org/schema/beans"
  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-2.5.xsd">

  <bean id="customerService" class="com.mkyong.customer.services.CustomerService"
    init-method="initIt" destroy-method="cleanUp">

    <property name="message" value="i'm property message" />
  </bean>

</beans>
```

Run it

```
package com.mkyong.common;

import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.mkyong.customer.services.CustomerService;

public class App
{
    public static void main( String[] args )
    {
        ConfigurableApplicationContext context =
            new ClassPathXmlApplicationContext(new String[] {"Spring-Customer.xml"});

        CustomerService cust = (CustomerService)context.getBean("customerService");

        System.out.println(cust);

        context.close();
    }
}
```

The `ConfigurableApplicationContext.close` will close the application context, releasing all resources and destroying all cached singleton beans.

Output

```
Init method after properties are set : i'm property message
com.mkyong.customer.services.CustomerService@47393f
...
INFO: Destroying singletons in org.springframework.beans.factory.
support.DefaultListableBeanFactory@77158a:
defining beans [customerService]; root of factory hierarchy
Spring Container is destroy! Customer clean up
```

The **initIt()** method is called, after the message property is set, and the **cleanUp()** method is called after the `context.close()`;

Thoughts...

It's always recommended to use **init-method** and **destroy-method** in bean configuration file, instead of implement the InitializingBean and DisposableBean interface to cause unnecessarily coupled your code to Spring.

Download Source Code

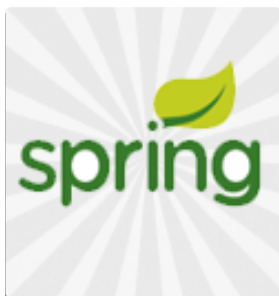
Download It – Spring-init-method-destroy-method-Example.zip
 (http://www.mkyong.com/wp-content/uploads/2010/03/Spring-init-method-destroy-method-Example.zip)

Tags : [spring \(http://www.mkyong.com/tag/spring/\)](http://www.mkyong.com/tag/spring/)

Share this article on

Twitter ([https://twitter.com/intent/tweet?text=Spring init-method and destroy-method example&url=http://www.mkyong.com/spring/spring-init-method-and-destroy-method-example/&via=mkyong](https://twitter.com/intent/tweet?text=Spring+init-method+and+destroy-method+example&url=http://www.mkyong.com/spring/spring-init-method-and-destroy-method-example/&via=mkyong)) Facebook (<https://www.facebook.com/sharer/sharer.php?u=http://www.mkyong.com/spring/spring-init-method-and-destroy-method-example/>) Google+ (<https://plus.google.com/share?url=http://www.mkyong.com/spring/spring-init-method-and-destroy-method-example/>)

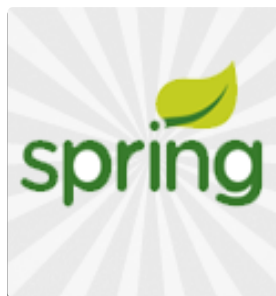
Reader also read :



Spring – Mixing XML and JavaConfig
 (http://www.mkyong.com/spring/spring-mixing-xml-and-javaconfig/)



Spring embedded database examples
 (http://www.mkyong.com/spring/spring-embedded-database-examples/)



Spring MethodInvokingFactoryBean Example
 (http://www.mkyong.com/spring/spring-methodinvokingfactorybean-example/)



Spring @Value – Import a list from properties file
 (http://www.mkyong.com/spring/spring-value-import-a-list-from-properties-file/)