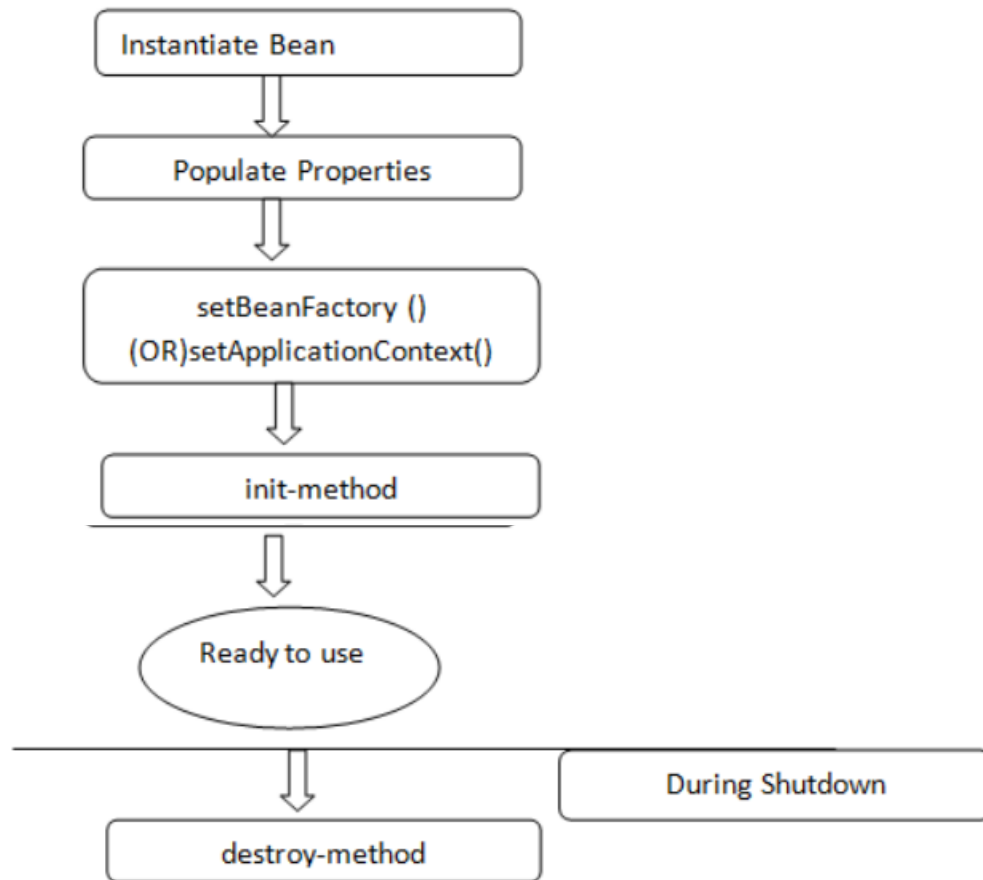


## Spring bean life cycle

### ➤ Spring bean life cycle steps

1. The containers instantiate every bean, populate their properties, and call the init method, which makes the bean ready to be used. During the shutdown of the application, the destroy method of the bean is invoked.
2. The init and destroy methods are used to perform initialization and cleanup operations.
3. Below diagram shows the Spring Bean life cycle.



### ➤ Example:

1. Create the class which implements

ApplicationContextAware, InitializingBean, DisposableBean interfaces.

```

10 package com.javaziearn.core;
11
12 import org.springframework.beans.BeansException;
13 import org.springframework.beans.factory.DisposableBean;
14 import org.springframework.beans.factory.InitializingBean;
15 import org.springframework.context.ApplicationContext;
16 import org.springframework.context.ApplicationContextAware;

```

```

public class SpringBeanDemo implements
ApplicationContextAware, InitializingBean, DisposableBean {

```

```

private String message;

```

```

23
24     public SpringBeanDemo() {
25         System.out.println("SpringBeanDemo is instantiating.");
26     }
27
28     public void setMessage(String message) {
29         this.message = message;
30         System.out.println("Inside setMessage method");
31     }
32
33     public String getMessage() {
34         return message;
35     }
36
37     @Override
38     public void destroy() throws Exception {
39         System.out.println("Inside destroy method");
40     }
41
42
43     @Override
44     public void afterPropertiesSet() throws Exception {
45         System.out.println("Inside afterPropertiesSet method");
46     }
47
48
49     @Override
50     public void setApplicationContext(ApplicationContext
51     applicationContext)throws BeansException {
52         System.out.println("Inside setApplicationContext method:");
53     }
54 }

```

## 2.Create the Spring configuration file

```

10 <beans xmlns="http://www.springframework.org/schema/beans"
11     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
12     xsi:schemaLocation="http://www.springframework.org/schema/beans
13     http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">
14
15     <bean id="springBean" class="com.java2learn.core.SpringBeanDemo">
16         <property name="message" value="Hello Spring Bean!!"></property>
17     </bean>
18
19 </beans>

```

## 3. Create the client and run it.

```

10 package com.java2learn.client;
11
12 import org.springframework.context.ConfigurableApplicationContext;
13 import org.springframework.context.support.FileSystemXmlApplicationContext;

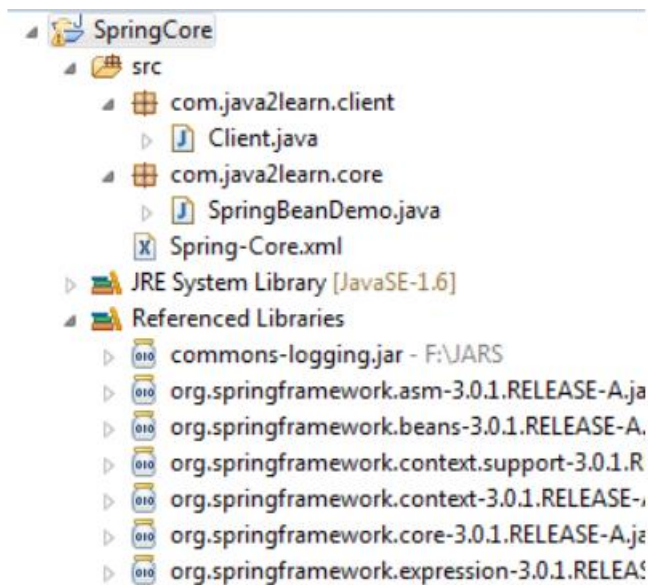
```

```

15 | import com.java2learn.core.SpringBeanDemo;
16 |
17 | public class Client {
18 |
19 |     public static void main(String[] args) {
        ConfigurableApplicationContext applicationContext = new
        FileSystemXmlApplicationContext("classpath:Spring-Core.xml");
        SpringBeanDemo bean = (SpringBeanDemo)
        applicationContext.getBean("springBean");
        System.out.println("message ::" + bean.getMessage());
        // shutdown the context
        applicationContext.close();
    }

```

#### 4. Final project structure.



#### ➤ Output:

```

SpringBeanDemo is instantiating.
Inside setMessage method
Inside setApplicationContext method:
Inside afterPropertiesSet method
message ::Hello Spring Bean!!
Inside destroy method

```