为了定义安装和拆卸一个 bean，我们只要声明带有 init-method 和/或 destroy-method 参数的 。init-method 属性指定一个方法，在实例化 bean时 ， 立即调用该方法。destroy-method 指定一个方法，只有从容器中移除 bean 之后，才调用该方法。
Bean的生命周期可以表达为：Bean的定义——Bean的初始化——Bean的使用——Bean的销毁

初始化回调

org.springframework.beans.factory.InitializingBean 接口指定一个单一的方法：

```
void afterPropertiesSet() throws Exception;
```

可以简单地实现上述接口和初始化工作可以在 afterPropertiesSet() 方法中执行，如下所示：

```
public class ExampleBean implements InitializingBean {
   public void afterPropertiesSet() {
      // do some initialization work
   }
}
```

在基于 XML 的配置元数据的情况下，可以使用 init-method 属性来指定带有 void 无参数方法的名称。例如：

```
<bean id="exampleBean"
        class="examples.ExampleBean" init-method="init"/>
```

下面是类的定义：

```
public class ExampleBean {
   public void init() {
      // do some initialization work
   }
}
```

销毁回调

org.springframework.beans.factory.DisposableBean 接口指定一个单一的方法：

```
void destroy() throws Exception;
```

因此，你可以简单地实现上述接口并且结束工作可以在 destroy() 方法中执行，如下所示：

```
public class ExampleBean implements DisposableBean {
   public void destroy() {
      // do some destruction work
   }
}
```

在基于 XML 的配置元数据的情况下，你可以使用 destroy-method 属性来指定带有 void 无参数方法的名称。例如：

```
<bean id="exampleBean"
        class="examples.ExampleBean" destroy-
method="destroy"/>
```

下面是类的定义：

```
public class ExampleBean {
   public void destroy() {
      // do some destruction work
   }
}
```