



链滴

# Spring 知识点总结

作者: [HuanYuanHe](#)

原文链接: <https://ld246.com/article/1573805579467>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 1.BeanFactory 和 ApplicationContext 的区别

BeanFactory 才是 Spring 容器中的顶层接口。ApplicationContext 是它的子接口。

BeanFactory 和 ApplicationContext 的区别：

1.创建对象的时间点不一样。

ApplicationContext：只要一读取配置文件，默认情况下就会创建对象。

BeanFactory：什么使用什么时候创建对象。

## 2.ApplicationContext 接口的实现类

ClassPathXmlApplicationContext：它是从类的根路径下加载配置文件 推荐使用这种

FileSystemXmlApplicationContext：它是从磁盘路径上加载配置文件，配置文件可以在磁盘的任意位置。

AnnotationConfigApplicationContext:当我们使用注解配置容器对象时，需要使用此类来创建 spring 容器。它用来读取注解。

## 3.bean 标签

作用：

用于配置对象让 spring 来创建的。

默认情况下它调用的是类中的无参构造函数。如果没有无参构造函数则不能创建成功。

属性：

id：给对象在容器中提供一个唯一标识。用于获取对象。

class：指定类的全限定类名。用于反射创建对象。默认情况下调用无参构造函数。

scope：指定对象的作用范围。

- singleton :默认值，单例的.
- prototype :多例的.
- request :WEB 项目中, Spring 创建一个 Bean 的对象, 将对象存入到 request 域中.
- session :WEB 项目中, Spring 创建一个 Bean 的对象, 将对象存入到 session 域中.
- global session :WEB 项目中, 应用在 Portlet 环境. 如果没有 Portlet 环境那么

globalSession 相当于 session.

init-method：指定类中的初始化方法名称。

destroy-method：指定类中销毁方法名称。

##4.bean标签生命周期

单例对象： scope="singleton"

一个应用只有一个对象的实例。它的作用范围就是整个引用。

生命周期：

对象出生：当应用加载，创建容器时，对象就被创建了。

对象活着：只要容器在，对象一直活着。

对象死亡：当应用卸载，销毁容器时，对象就被销毁了。

多例对象： scope="prototype"

每次访问对象时，都会重新创建对象实例。

生命周期：

对象出生：当使用对象时，创建新的对象实例。

对象活着：只要对象在使用中，就一直活着。

对象死亡：当对象长时间不用时，被 java 的垃圾回收器回收了。

##5.实例化 Bean 的三种方式

**第一种方式：**使用默认无参构造函数

在默认情况下：它会根据默认无参构造函数来创建类对象。如果 bean 中没有默认无参构造函数，将创建失败。

```
<bean id="accountService" class="com.itheima.service.impl.AccountServiceImpl"/>
```

**第二种方式：** spring 管理静态工厂-使用静态工厂的方法创建对象

```
/** * 模拟一个静态工厂，创建业务层实现类 */
public class StaticFactory { public static IAccountService createAccountService(){
return new AccountServiceImpl();
}
}
```

此种方式是使用 StaticFactory 类中的静态方法 createAccountService 创建对象，并存入 spring 容器

id 属性：指定 bean 的 id，用于从容器中获取

class 属性：指定静态工厂的全限定类名

factory-method 属性：指定生产对象的静态方法

```
<bean id="accountService" class="com.itheima.factory.StaticFactory" factory-method="creat
AccountService"> </bean>
```

**第三种方式：** spring 管理实例工厂-使用实例工厂的方法创建对象

```
/** * 模拟一个实例工厂，创建业务层实现类 * 此工厂创建对象，必须现有工厂实例对象，
```

```
再调用方法 使用某个类中的方法创建对象，并存入spring 容器 */
public class InstanceFactory {
    public IAccountService createAccountService(){
        return new AccountServiceImpl();
    }
}
```

此种方式是先把工厂的创建交给 spring 来管理。

然后在使用工厂的 bean 来调用里面的方法

factory-bean 属性：用于指定实例工厂 bean 的 id。

factory-method 属性：用于指定实例工厂中创建对象的方法。

```
<bean id="instancFactory" class="com.itheima.factory.InstanceFactory"></bean> <bean id=
accountService" factory-bean="instancFactory" factory-method="createAccountService"></b
an>
```