spring学习笔记—bean介绍

|  |  |
| --- | --- |
|  | 🢂 内容概览 |
|  | Why：此文档用来做什么？它存在的意义是什么？为解决什么问题？   |  | | --- | |  |   What：当前包含了那些内容？   |  | | --- | |  |   How：此文档应如何参考？   |  | | --- | |  |   Who：此文档适用于那些人员阅读参考？   |  | | --- | |  | |

目录

[1 Bean管理简介（BeanDefination） 3](#_Toc478740836)

[1.1 Ioc容器如何描述Bean? 3](#_Toc478740837)

[1.2 如何将外部对象注册到Ioc中？ 3](#_Toc478740838)

[1.3 Bean的命名规则是怎样的？ 4](#_Toc478740839)

[1.4 初始化bean有哪些方式？ 4](#_Toc478740840)

# Bean管理简介（BeanDefination）

## Ioc容器如何描述Bean?

容器使用BeanDefination描述Bean的定义，其中包含很多元数据信息；元数据信息包括如下：

1. 类名：Bean实现类的类名；
2. 状态信息：Bean状态相关的信息，如scope，生命周期回调函数等等；
3. 依赖关系：对其它Bean的引用；
4. 其它信息：比如连接池大小等；

元数据信息以属性的形式存放在BeanDefination对象中，用来组成Bean的定义；属性如下：

|  |  |
| --- | --- |
| **Property** | **Explained in...** |
| class | the section called “Instantiating beans” |
| name | the section called “Naming beans” |
| scope | Section 6.5, “Bean scopes” |
| constructor arguments | the section called “Dependency Injection” |
| properties | the section called “Dependency Injection” |
| autowiring mode | the section called “Autowiring collaborators” |
| lazy-initialization mode | the section called “Lazy-initialized beans” |
| initialization method | the section called “Initialization callbacks” |
| destruction method | the section called “Destruction callbacks” |

## 如何将外部对象注册到Ioc中？

IOC容器能够注册外部创建的对象到容器中；通过使用ApplicationContext的getBeanFactory()方法获取BeanFactory对象来进行注册；BeanFactory默认使用DefaultListableBeanFactory作为实现类，该类支持通过registerSingleton()和registerBeanDefinition()方法注册Bean；

注：

1. 注册外部Bean到容器中，需要尽早调用上述两个方法；以便于容器能够正确的进行自动注入和其它自省操作；
2. 能够通过注册，重写Bean的定义，但该操作并非线程安全的，可能引起并发异常或者状态的不一致；

## Bean的命名规则是怎样的？

1. 每个Bean可以有一个或者多个标识符，标识符在容器中必须是唯一的；一个bean通常只有一个标识符，可以通过name属性或者alias标签定义多个别名；
2. ID属性：使用唯一的字符串定义bean；
3. Name属性：定义bean的别名，多个别名通过逗号或者分号分隔；
4. IOC容器默认提供bean的ID，默认约定以驼峰命名法命名；
5. 通过alias可以定义别名；

|  |
| --- |
| **<alias name**=**"fromName" alias**=**"toName"/>** |

1. 通过@Bean注解定义ID或者别名；
2. 静态内部类定义，需要指定外部类名字；如：com.example.Foo$Bar；

## 初始化bean有哪些方式？

1. 初始化Bean包括：构造器初始化，静态工厂方法，实例工厂方法；
2. 构造器初始化：IOC容器会通过调用bean的构造器来初始化Bean实例；
3. 静态工厂方法初始化：通过调用指定的静态工厂方法，获得bean实例；静态工厂方法需要以需求自己实现；同一个工厂可以返回不同类型的bean；

|  |
| --- |
|  |

1. 实例工厂方法：通过factory-bean和factory-method方法直接指定其它bean作为工厂，初始化当前bean；可以通过同一个工厂的不同方法初始化多个bean；

|  |
| --- |
|  |