

第八章：Spring Bean 作用域

小马哥 • mercyblitz

Spring Bean 作用域

1. Spring Bean作用域
2. "singleton" Bean作用域
3. "prototype" Bean作用域
4. "request" Bean作用域
5. "session" Bean作用域
6. "application" Bean作用域
7. 自定义Bean作用域
8. 课外资料
9. 面试题精选



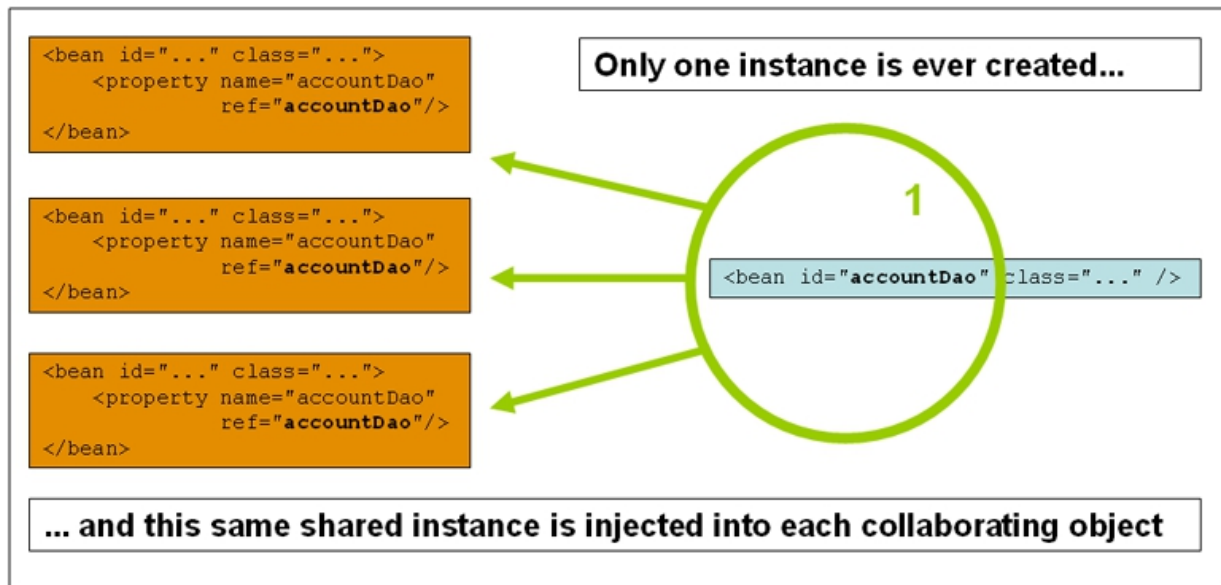
Spring Bean作用域

- 作用域

来源	说明
singleton	默认 Spring Bean 作用域，一个 BeanFactory 有且仅有一个实例
prototype	原型作用域，每次依赖查找和依赖注入生成新 Bean 对象
request	将 Spring Bean 存储在 ServletRequest 上下文中
session	将 Spring Bean 存储在 HttpSession 中
application	将 Spring Bean 存储在 ServletContext 中

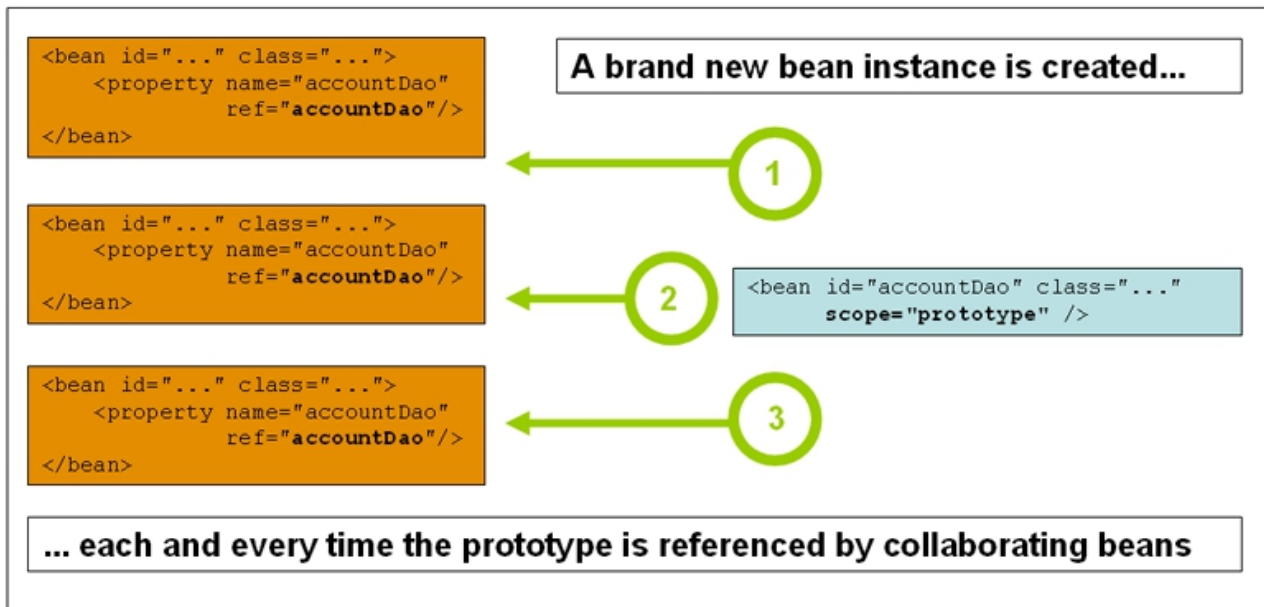
“singleton” Bean作用域

- 配置



“prototype” Bean作用域

- 配置



“prototype” Bean作用域

- 注意事项
 - Spring 容器没有办法管理 prototype Bean 的完整生命周期，也没有办法记录实例的存在。销毁回调方法将不会执行，可以利用 BeanPostProcessor 进行清扫工作。

“request” Bean作用域

- 配置

- XML - `<bean class="..." scope = "request" />`

- Java 注解 - `@RequestScope` 或 `@Scope(WebApplicationContext.SCOPE_REQUEST)`

- 实现

- API - `RequestScope`

“session” Bean作用域

- 配置

- XML - `<bean class="..." scope = "session" />`

- Java 注解 - `@SessionScope` 或 `@Scope(WebApplicationContext.SCOPE_SESSION)`

- 实现

- API - `SessionScope`

“application” Bean作用域

- 配置

- XML - `<bean class="..." scope = "application" />`
- Java 注解 - `@ApplicationScope` 或 `@Scope(WebApplicationContext.SCOPE_APPLICATION)`

- 实现

- API - `ServletContextScope`

自定义 Bean 作用域

- 实现 Scope
 - `org.springframework.beans.factory.config.Scope`
- 注册 Scope
 - API - `org.springframework.beans.factory.config.ConfigurableBeanFactory#registerScope`
 - 配置

```
<bean class="org.springframework.beans.factory.config.CustomScopeConfigurer">  
    <property name="scopes">  
        <map>  
            <entry key="...">  
            </entry>  
        </map>  
    </property>  
</bean>
```

课外资料

- Spring Cloud RefreshScope是如何控制Bean的动态刷新？

面试题

沙雕面试题 - Spring 内建的 Bean 作用域有几种？

答：singleton、prototype、request、session、application 以及 websocket



我真的没笑

面试题

996 面试题 - singleton Bean 是否在一个应用是唯一的‘



答：否则，singleton bean 仅在当前 Spring IoC 容器（BeanFactory）中
是单例对象。

面试题

劝退面试题 - “application” Bean 是否被其他方案替代



答：可以的，实际上，“application” Bean 与 “singleton” Bean 没有本质区别