

# Spring容器

2021年9月19日 14:34

Spring容器是Spring核心的部分，主要作用是帮助程序员处理大量繁琐事务，使程序员只需要用代码去实现他所关注的业务，而不需要去处理其他的事务

在Spring的框架中，程序员是不需要自己去创建对象的，也不需要去创建对象之间的关联，这些信息只需要通过**配置文件**去告诉Spring容器，**由Spring容器去负责创建对象，去建立对象之间的关联**，这些对象被称为**Bean对象**

Spring容器不仅能创建对象、创建对象之间的关联，还能够管理对象的生命周期，从创建到销毁

## 三种写配置信息的方式

### 1. XML文件

好处：配置信息与代码分离，是一个放在外部的独立文件，可以直接修改配置而不编译代码

弊病：XML配置比较繁琐；配置信息与代码分离，无法及时检查出XML配置中的错误，只有当程序运行到使用配置的这一部分时，才能发现配置的错误

### 2. Java代码

用Java写配置可以提高类型的安全性，方便重构

### 3. 注解

基于注解的自动配置大幅度减少了配置的工作量

不论是哪种方式，都需要写大量的配置，所以Spring提供了**SpringBoot**，大量采用默认配置，减少写配置的工作量，帮助开发者高效的构建Spring应用。需要写的只是那些**特别的**，不能用默认去表示的配置

## 用注解来配置Bean对象的一些例子

### @Component

在类前加一个注解**@Component**，Spring容器在读到这样一个注解时，就会知道这个类是需要**创建出来的一个对象**

注解一般有两种属性：**id**和**scope**

id：Bean对象的名称，理论上可以任意命名，默认为**类名首字母小写**

scope：指定Bean对象的作用域，默认为**singleton**，表示在Spring容器中只会存在一个该Bean

- `@Component("customerController")`
- `@Scope("prototype")`

```

6  /**
7   * @author: Ming Qiu
8   * @date: Created in 17:01 2020/7/31
9   */
10 @Component("boss_Setter")
11 public class Boss_Setter {
12
13     private Car car;
14     private Office office;
15
16     @Autowired
17     public void setCar(Car toyota) { this.car = toyota; }
18
19
20
21     @Autowired
22     public void setOffice(Office office) { this.office = office; }
23
24
25
26 }
27

```

## 其他层的注解举例

@Controller——告诉Spring容器这是一个控制器层（Controller层）的Bean对象

@Service——告诉Spring容器这是一个服务层（Service层）的Bean对象

@Repository——告诉Spring容器这是一个数据访问层（Dao层）的Bean对象

@Mapper——告诉Spring容器这是一个映射层（Mapper层）的Bean对象

## Spring框架的两种容器

### 1. BeanFactory

BeanFactory是Spring框架最基本的容器，创建对象、创建对象关联、实例化对象、销毁对象等事情都是BeanFactory来完成的

BeanFactory的逻辑是，当它第一次访问到一个对象时，它才会创建对象

### 2. ApplicationContext

ApplicationContext由BeanFactory派生而来的容器，具备BeanFactory的所有特性，同时也增加了高级功能，如：国际化资源接口MessageSource、资源加载接口ResourceLoader、应用事件发布接口ApplicationEventPublisher

ApplicationContext的逻辑是在容器启动时就会创建所有的对象