

## Spring容器的声明和开启注解的扫描

Spring中，要有一个IoC容器，作为大本营，如果使用注解的话，要为这个容器开启对于注解的支持和对组件的自动扫描。

如果是xml文件：xml文件本身就是一个容器，如果要开启注解的支持

<context:annotation-config>:**注解扫描是针对已经在Spring容器里注册过的Bean**

<context:component-scan>:不仅具备<context:annotation-config>的所有功能，还可以在指定的package下面扫描对应的bean

其中可以设置具体的扫描路径

在Java代码中，使用配置类，在类上使用标注@Configuration，表明这是一个配置类。再添加注解@CompomentScan（basepackage = ““）填写基本扫描路径。默认的路径是标注这个类所在的包路径。

Springboot项目中，帮我们自动生成了启动类启动类上的注解@SpringBootApplication:包含三方面@SpringBootConfigruation(里面包含配置注解@Configuration)，包含@ComponentScan(默认的扫描路径是启动类所在的包及子包)，还包含@Import和@EnableAutoConfiguration

为我们配置好了上下文，bean的扫描路径为所在的包中。还有就是导入的东西。

**重要：@Configuration**

一个spring中可以有多个@Configuratoin

1. @Configuration是个组件，存在于IoC容器中
2. @Configuration有主次之分，主配置类是驱动整个程序的入口。可以是一个或多个（多个要用@Order进行排序）
3. 平时书写的@Configuration属于次配置类，属于一个spring的组件放在IoC容器中，主要是通过主配置类的@ComponentScan和@Import进行加载。
4. 配置类可以进行嵌套

## spring中的依赖关系

spring中的bean：1.将一个Java类作为一个bean对象实例化，并登记到IOC容器中。2. 一个Java类对象，属性是另一个Java类，依赖另一个Java类。建立这两个对象之间的依赖关系，叫做依赖注入DI，或者是装配（一个的注解使用在类中的方法，字段参数上的，一个注解是用在类上面的）两者的关系是，要是有自动装配注解，那么这个类要是一个java

bean，即包含在@Component类中。

**装配：自动装配**，xml中显示装配，JavaConfiguration中自动装配。

自动装配：基础：有一个配置类，在配置类上设置自动扫描@Configuration，@ComponentScan

方法：在Java类中的字段上，设置@Autowried。在Java类中的构造方法上，setter方法上，其他方法上，设置@Autowried。都会自动装配这些方法的参数所对应的Javabean

其他：@Autowried是通过byType的方法去寻找JavaBean的，如果同一类型有多个，会报错，解决方法：添加注解@Qualifier（“id”）,在这些相同类型中的Javabean中注入id这个bean

@Autowried（required= true）这是默认的，要求被注入的bean必须存在，不存在会报错。

@Autowried是spring中特有的注解，是通过byType在IOC容器中寻找bean进行匹配的。如果想要指定名称，要配合注解@Qualifier一起使用

@Resource是Java2EE里面的一个注解，和@Autowried的使用方法差不多，但是可以设置属性（name和type），表示按照名称和类型进行注入，并不需要配合@Qualifier来使用。默认是通过name来进行注入的。如果细分，可以分成@Repository，@Service，@Controller这个是一个细分

**显示装配**：在JavaConfig类中显示装配，在Java类的方法中设置@Bean（是在配置类当中使用的），就会将方法的返回类型注册为一个bean，同时如果这个方法中有对其他bean的依赖，比如构造方法，或者是setter方法，那么就会将方法所需要的参数自从注入进来。

**通过xml显示配置**：使用<constructor-arg ref = “compactDisc”>元素，通过setter方法，将字段注入，使用标签<property name=”字段名” value = “要注入的javabean“

为了简化书写，不写<constructor-arg ref = “”> 使用**c命名空间**。就可以写成

c: cd-ref=”compactDisc

同理**P命名空间**p: 参数-ref = ““，还可以把参数换成占位符

如果有集合可以使用list和set标签（set是无序不可重复的）

**注意，value=““表示的是字符串，而ref=”“表示的是引用。**

## spring中的bean的声明

使用了自动注入属性的类，就得声明为bean

注解中的@Component,@Respository, @Service, @Controller

Xml文件中的<bean />标签

注解类中和@Configuration配合使用的@Bean

### 消除自动注解的歧义

@Autowried和@Qualifier(填写id)配合使用，

在类上使用@Primary，表明这个bean是同类型中的首选bean

### 作用域

* 单例模式：都是同一个javabean（默认为单例模式）
* 原型模式：每次获取的都是新创建的javaBean
* Session：每次会话生成一个bean
* Request：为每次请求生成一个bean

### @Conditional(前置条件)

表明在其他bean创建后，才会创建。

## Java Persistence API（Java持久化api）

持久化，和数据库相关。Mybatis和jap框架就是把jdbc封装的持久化框架。

两个注解@Entity和@Table

@Entity表明这个类是个实体类，使用默认的ORM规则：类名对应数据库中的表名，字段名对应表中的字段。

在spring data jpa中才能使用，在mybatis中没有

1. 实体类要使用注解@Entity，：org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'entityManagerFactory' defined in class path resource [org/springframework/boot/autoconfigure/orm/jpa/HibernateJpaAutoConfiguration.class]: Invocation of init method failed; nested exception is org.hibernate.AnnotationException: @OneToOne or @ManyToOne 异常

用@Entity标注的实体类，表的名字和类的名字相同，如果不同，则添加属性@Entity（name=“表名“）。或者加上注解@Table（”name=表名“）同时@Table（catalog = ”数据库名“）

1. 实体类中要有一个主键，用注解@Id标明。和主键一起使用的注解@GeneratedValue(strategy = GeneratedType.IDENTITY)表明主键自增。这是个枚举类。
2. @Column: 表明数据库中列的属性，unque是否唯一，nullable是否为空等等。
3. 必须拥有一个默认的无参构造方法。
4. 这个实体类不能是枚举类，和接口，也不能被fianl修饰。