

装载Bean3

自动装载Bean（组件扫描[component scanning]）

- Spring 能够从 classpath 下自动扫描, 侦测和实例化具有特定注解的组件.
- 特定组件包括:
 1. @Component: 基本注解, 标识了一个受 Spring 管理的组件
 2. @Repository: 标识持久层组件
 3. @Service: 标识服务层(业务层)组件
 4. @Controller: 标识表现层组件
- 对于扫描到的组件, Spring 有默认的命名策略: 使用非限定类名, 第一个字母小写. 也可以在注解中通过 value 属性值标识组件的名称
- 当在组件类上使用了特定的注解之后, 还需要在 Spring 的配置文件中声明 context:component-scan :
 1. base-package 属性指定一个需要扫描的基类包, Spring 容器将会扫描这个基类包里及其子包中的所有类.
 2. 当需要扫描多个包时, 可以使用逗号分隔.
- 如果仅希望扫描特定的类而非基包下的所有类, 可使用 resource-pattern 属性过滤特定的类, 示例:

1. context:include-filter 子节点表示要包含的目标类
2. context:exclude-filter 子节点表示要排除在外的目标类
3. context:component-scan 下可以拥有若干个 context:include-filter 和 context:exclude-filter 子节点, 使用 context:include-filter 的时候需要将 use-default-filters 属性设置成 false, 否则无效!

```
<context:component-scan base-package="aop.yxd.server,aop.yxd.aspect"></
  <context:include-filter type="annotation" expression="org.springframework.stereotype.Controller" />
  <context:exclude-filter type="aspectj" expression="aop.yxd.aspect.LoggingAspect"/>
</context:component-scan>
```

- component-scan 还可以自动装配具有 @Autowired 和 @Resource、@Inject 注解的属性. 当未使用组件扫描器, 需要添加注解的自动注入的扫描器的时候. 方可使用上面的注解方式的注入。

```
<context:annotation-config>
```

PropertyPlaceholderConfigurer 配置文件的注入

spring2.5之前需要配置一个spring的内部bean, 去关联你的properties配置文件。2.5之后简化了这一步骤。

```
<context:property-placeholder location="classpath:db.properties" />
```

标签内部还可以再配置其他的属性, 如果与properties文件中同名, 则以标签内部为准, 同时内部的优先级高于文件中属性配置。

spEl

自己可以去学, 意义不到, 就不总结了。