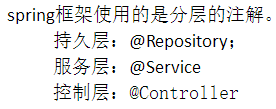
Spring的IoC创建注解方式详解

# 依赖注入的注解方式

我们可以给我们的类加上**spring组件注解**，只需再配置下**spring的扫描器**就可以实现**bean的自动载入**。



**@Component（组件）：**

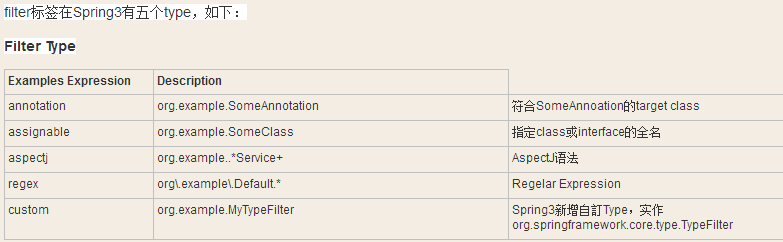
**@Service（服务层）@Controller（控制层）@Repository(数据库访问层)**

有了<context:component-scan>，另一个<context:annotation-config/>标签根本可以移除掉，因为已经被包含进去了。

<context:component-scan>提供两个子标签：**<context:include-filter>和<context:exclude-filter>各代表引入和排除的过滤。**

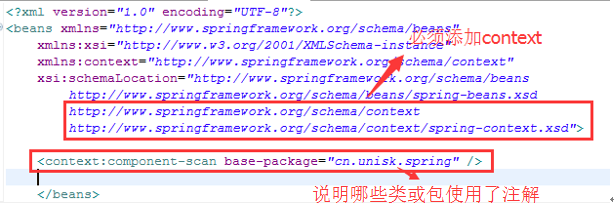
<context:component-scan base-package="com.xhlx.finance.budget" >

**<context:include-filter type="regex" expression=".service.\*"/>  
</context:component-scan>**



# 使用注解开发的步骤

## 在配置文件中声明context，并指出哪些类或包使用了注解。



## 在相应的类中使用注解就可以了。

# 问题解决

org.springframework.beans.factory.BeanDefinitionStoreException: Failed to read candidate component class: file [E:\JAVA\EclipseWorkSpace\SpringTest\bin\cn\unisk\spring\AppTest.class]; nested exception is org.springframework.core.NestedIOException: ASM ClassReader failed to parse class file - probably due to a new Java class file version that isn't supported yet: file [E:\JAVA\EclipseWorkSpace\SpringTest\bin\cn\unisk\spring\AppTest.class]; nested exception is java.lang.IllegalArgumentException

at org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider.findCandidateComponents(ClassPathScanningCandidateComponentProvider.java:290)

问题：**当前Spring与JDK不兼容，换用JDK1.6或JDK1.7即可。**

**google问题，一下子解决。**

<http://blog.csdn.net/sihai12345/article/details/54692303?locationNum=7&fps=1>

# 例子

ApplicationContext.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

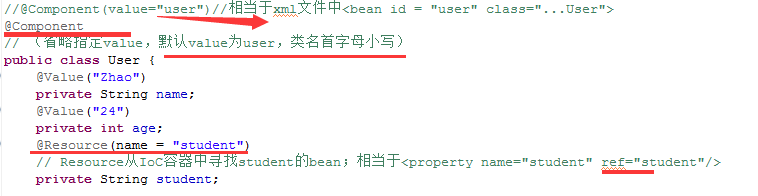
*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd"*>

**<context:component-scan base-package=*"cn.unisk.spring"*/>**

</beans>



# Component、Service、Controller、Repository介绍

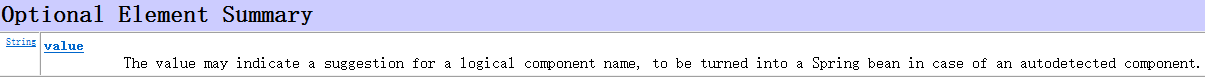
## Component只需要指定一个属性：value。

作用：Indicates that an annotated class is a "component".Such classes are considered as candidates for auto-detection when using annotation-based configuration and classpath scanning.



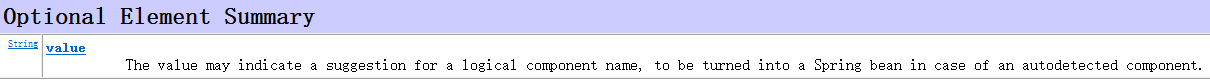
## Repository：只需要指定一个属性value。（DAO）指定该类是一个DAO组件。持久层

作用：Indicates that an annotated class is a "Repository" (or "DAO").

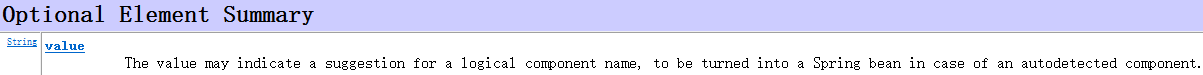


## Service业务逻辑层

作用：Indicates that an annotated class is a "Service" (e.g. a business service facade).



## Controller 控制层



作用：Indicates that an annotated class is a "Controller" (e.g. a web controller).

## 总结：

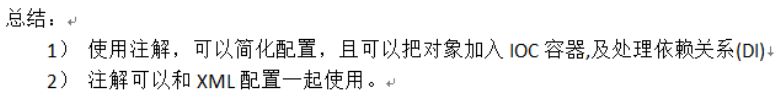
相同点：都是只有一个属性，value，可以指定，也可以不指定，采用默认值。

不同点：对于任何一个类都可以指定Component，但是利用Service、Controller、Repository更能说明该类的作用。

# 注解方法与XML配置的比较

**注解方式与XML配置可以同时结合使用。**

注解方式方便操作，但是不利于控制。



spring框架为我们提供了注解功能。

使用注解编程，主要是为了替代xml文件，提高开发效率。但是，xml文件的使用就是解决修改程序的弊端，现在又不去使用xml文件，那么不就违背了开闭原则了么，的确是。不过么，注解也有注解的好，使用注解就不用配置那么多的xml文件啦，最重要的是开发效率高。 在没有使用注解时，spring框架的配置文件**applicationContext.xml**文件中需要配置很多的**<bean>标签**，用来声明类对象。使用注解，则不必在配置文件中添加标签拉，对应的是**在对应类的“注释”位置添加说明**。

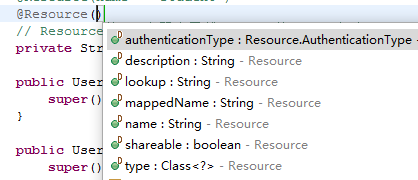
# @Resource解析

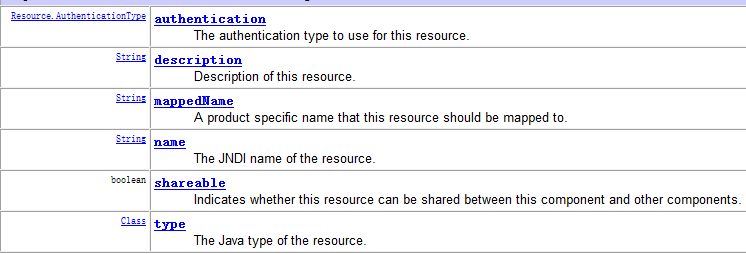


作用：**Resource 注释标记应用程序所需的资源**。此注释可以应用于应用程序组件类，或者该组件类的字段或方法。如果将该注释应用于一个字段或方法，那么初始化应用程序组件时容器将把所请求资源的一个实例注入其中。如果将该注释应用于组件类，则该注释将声明一个应用程序在运行时将查找的资源。

**Spring中注解所用的Resource是javax.annotation包中的Resource**。

Resource是一个注解，属性有6个，最常用的就是name，通过name从IoC中寻找Bean对象。也可以不指定name，默认情况下，首先使用的是ByName，（**默认名字**），如果根据名字查找不到关联对象，然后再采用的是ByType通过类型从IoC中获取**Bean对象**，这种方式要求IoC中只有一个该Bean对象，否则容易出错。所以建议采用ByName获取Bean对象。





# 注解：

