1.1 Spring的IOC的注解开发

## 1.1.1 spring的注解开发的入门

### 1.1.1.1 创建项目并引入相关jar包

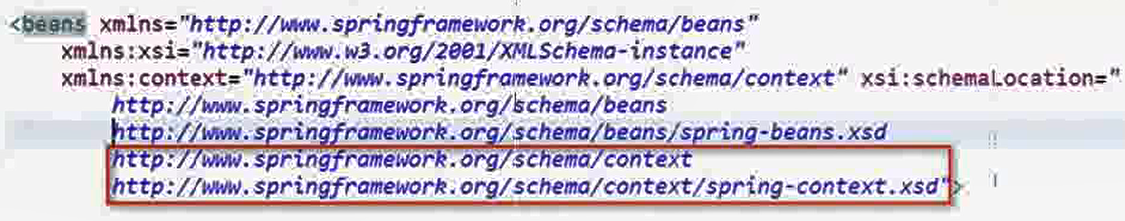
\* 在spring4的版本中,除了引入基本的jar包外还需要引入aop包。



### 1.1.1.2 引入Spring的配置文件

\* 在资源文件夹下创建applicationContext.xml文件

a) 引入约束：使用注解开发需要引入context约束(xsd-configuration.html)



### 1.1.1.3 创建接口和实现类

### 1.1.1.4 IOC注解的使用

a) 添加组件扫描



b) 在类上面添加注解



### 1.1.1.5 注解方式设置属性的值

·注解方式：使用注解方式,可以没有set方法

1. 如果属性没有set方法,直接在属性上添加注解即可
2. 如果属性有set方法,则需要在set方法上添加注解

C:\Users\23785\AppData\Local\Temp\1551952399(1).png

## 1.1.2 Spring的IOC的注解的详解

### 1.1.2.1 @component：组件

·修饰一个类,将这个类交给spring管理

·这个注解有3个衍生注解(功能类似)、修饰类的

1. @Controller : 修饰web层类
2. @Service ：修饰service层类
3. @Repository ：修饰dao层类

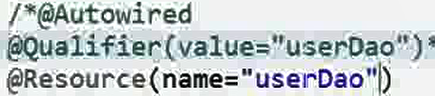
### 1.1.2.2 属性注入的注解

·普通属性：@Value ： 设置普通属性的值

·对象属性：1. @Autowired ：设置对象属性的值、但是按照属性的类型来完成注入的

因为我们习惯使按照名称完成注入：必须让@Autowired和@Qualifier一起用来完成名称注入

2. @Resource ：完成对象属性注入,按照名称完成属性注入



### 1.1.2.3 Bean的其他属性注解

1. 生命周期相关注解(了解)

· @PostConstruct : 初始化方法

· @PreDestroy ：销毁方法

2．Bean的作用范围的注解

· @Scope ：作用范围

取值：

1. singleton :默认单例
2. prototype ：多例
3. request 示例：C:\Users\23785\AppData\Local\Temp\1551954439(1).png
4. session
5. globalsession

## 1.1.3 IOC 的注解和XML开发的比较

### 1.1.3.1 XML和注解的比较

1. 适用场景：

a) XMl : 可以适用任何场景(类)

· 结构清晰、维护方便

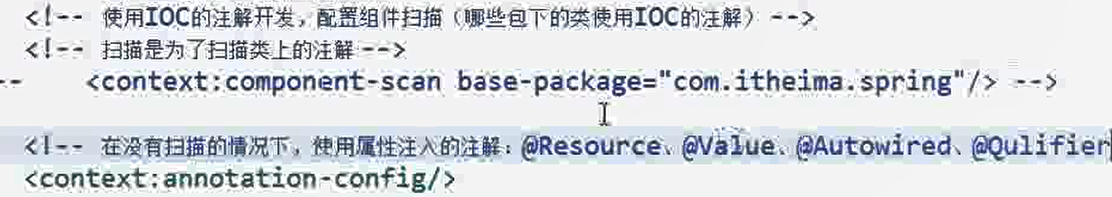
1. 注解 ：有些地方用不了,这个类不是自己提供的

·开发方便



### 1.1.3.2 XML和注解整合开发

·XML管理Bean、注解完成属性注入。



1.2 Spring的AOP的XML开发

## 1.2.1 AOP的概述

### 1.2.1.1 什么使AOP

· AOP(面向切面编程)：通过预编译方式和运行期动态代理实现程序功能的统一和维护的一种技术。AOP是OOP的扩展和延申,用于解决OOP的问题

### 1.2.1.2 Spring底层的AOP实现原理

·动态代理

1. JDK动态代理 ：只能对于实现了接口的类产生代理
2. Cglib动态代理：类似于javassist第三方代理技术、对没有实现接口的类产生代理对象。生成子类对象。