

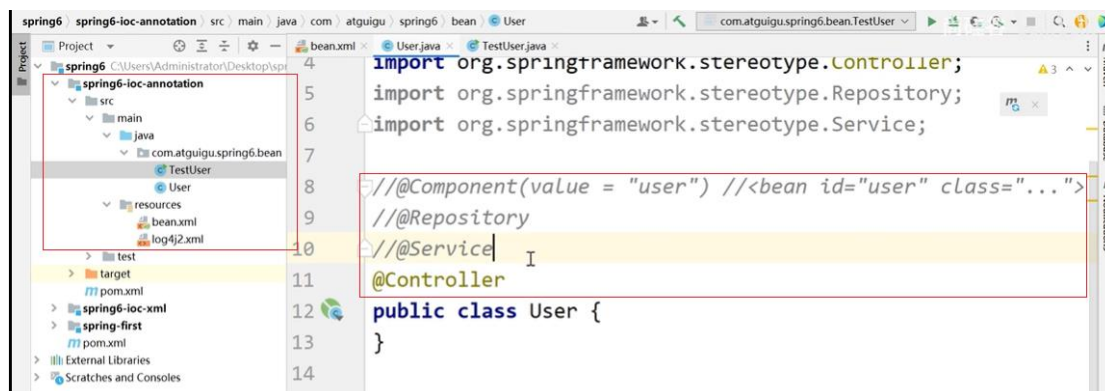
-----实际开发中基本是基于注解实现的-----

Spring 通过注解实现自动装配的步骤如下:

- 1、引入依赖
- 2、开启组件扫描
- 3、使用注解定义 Bean
- 4、依赖注入

环境准备:

依赖在父 pom.xml 已经引入, 需要复制 log4j2 和新建 bean.xml



开启组件扫描:

```
spring6 - bean.xml [spring6-ioc-annotation] - Administrator
spring6 - spring6-ioc-annotation - src - main - resources - bean.xml
xmlns:context= http://www.springframework.org/schema/context
xsi:schemaLocation="http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

<!-- 开启组件扫描-->
<context:component-scan base-package="com.atguigu.spring6"></context:component-scan>

</beans>
```

使用注解定义 bean:

```
spring6 - spring6-ioc-annotation - src - main - java - com - atguigu - spring6 - bean - User
com.atguigu.spring6.bean.TestUser
import org.springframework.stereotype.Controller;
import org.springframework.stereotype.Repository;
import org.springframework.stereotype.Service;

//@Component(value = "user") //<bean id="user" class="...">
//@Repository 可以不用写这个user 原始写法
//@Service
@Controller

public class User {
}
```

3.3.3、使用注解定义 Bean

Spring 提供了以下多个注解，这些注解可以直接标注在 Java 类上，将它们定义成 Spring Bean。

注解	说明
@Component	该注解用于描述 Spring 中的 Bean，它是一个泛化的概念，仅仅表示容器中的一个组件（Bean），并且可以作用在应用的任何层次，例如 Service 层、Dao 层等。使用时只需将该注解标注在相应类上即可。
@Repository	该注解用于将数据访问层（Dao 层）的类标识为 Spring 中的 Bean，其功能与 @Component 相同。
@Service	该注解通常作用在业务层（Service 层），用于将业务层的类标识为 Spring 中的 Bean，其功能与 @Component 相同。
@Controller	该注解通常作用在控制层（如 SpringMVC 的 Controller），用于将控制层的类标识为 Spring 中的 Bean，其功能与 @Component 相同。

测试:

