Spring – day02

学习目标

1. 能够使用jdbcTemplate完成增删改查操作
2. 能够了解基于xml配置声明式事务管理
3. 能够掌握基于annotation声明式事务管理
4. 能够了解spring5新特性

**学习指南**

使用jdbcTemplate完成增删改查操作

了解基于xml配置声明式事务管理（重点）

掌握基于annotation声明式事务管理（重点）

了解spring5新特性

# 第一堂课

**本节知识点**

1.1 今日课程内容介绍

1.2 JdbcTemplate的概述和入门

1.3 JdbcTemplate在spring的ioc中使用

**本节目标**

能够编写jdbcTemplate的入门程序

能够使用ioc管理上入门程序中的代码

**视频时间：**

17分12秒

## 今日课程内容介绍

### 知识概述

介绍今日课程安排

### 视频详情



### 总结与补充

1、spring中的JdbcTemplate

JdbcTemplate的作用：

它就是用于和数据库交互的，实现对表的CRUD操作

如何创建该对象：

对象中的常用方法：

2、作业：

spring基于AOP的事务控制

3、spring中的事务控制

基于XML的

基于注解的

### 课堂提问与练习

无

### 习题答案

无

## JdbcTemplate的概述和入门

### 知识概述

介绍JdbcTemplate，创建项目，做入门测试

### 视频详情



### 总结与补充

**JdbcTmplate概述**

它是 spring 框架中提供的一个对象，是对原始 Jdbc API 对象的简单封装。spring 框架为我们提供了很多的操作模板类。

操作关系型数据的：

JdbcTemplate

HibernateTemplate

操作 nosql 数据库的：

RedisTemplate

操作消息队列的：

JmsTemplate

我们今天的主角在 spring-jdbc-5.0.2.RELEASE.jar 中，我们在导包的时候，除了要导入这个 jar 包外，还需要导入一个 spring-tx-5.0.2.RELEASE.jar（它是和事务相关的）

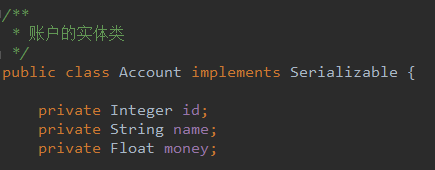
**创建项目**

day04\_eesy\_01jdbctemplate

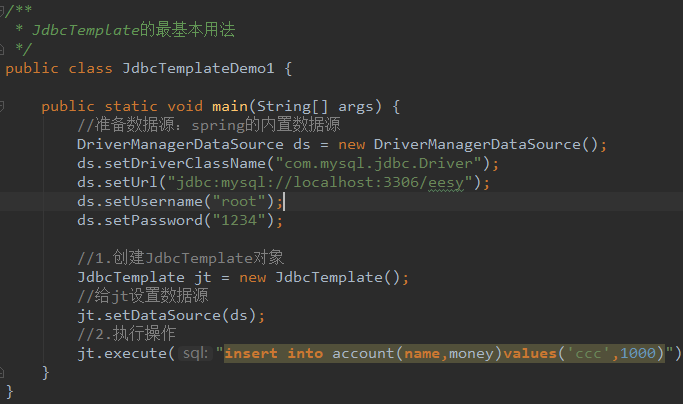
**pom文件**



**创建类**



**测试**



### 课堂提问与练习

无

### 习题答案

无

## JdbcTemplate在spring的ioc中使用

### 知识概述

把上一个测试类，使用springIOC的方式管理

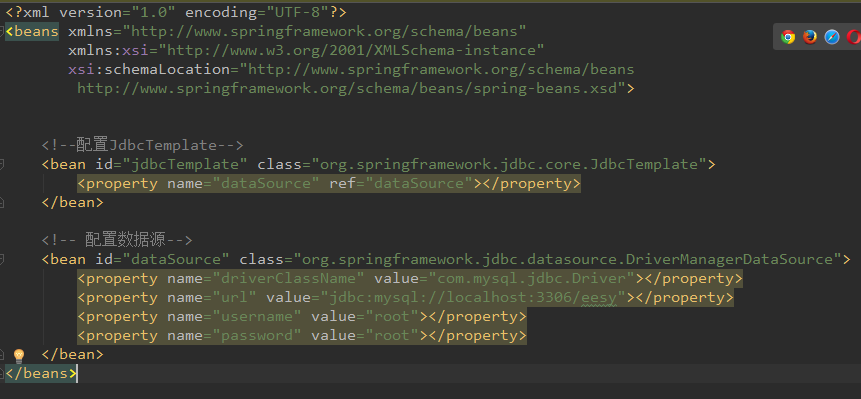
### 视频详情



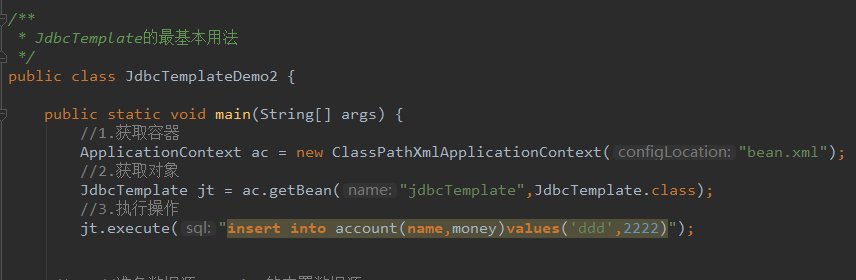
### 总结与补充

修改上一个测试的代码

创建bean.xml文件



创建测试类：



### 课堂提问与练习

无

### 习题答案

无

# 第二堂课

**本节知识点**

2.1 JdbcTemplate的CRUD操作

2.2 JdbcTemplate在Dao中的使用

**本节目标**

能够掌握JdbcTemplate中的CRUD操作

JdbcTemplate在Dao中的使用

**视频时间：**

31分34秒

## JdbcTemplate的CRUD操作

### 知识概述

使用JdbcTemplate进行CRUD的操作

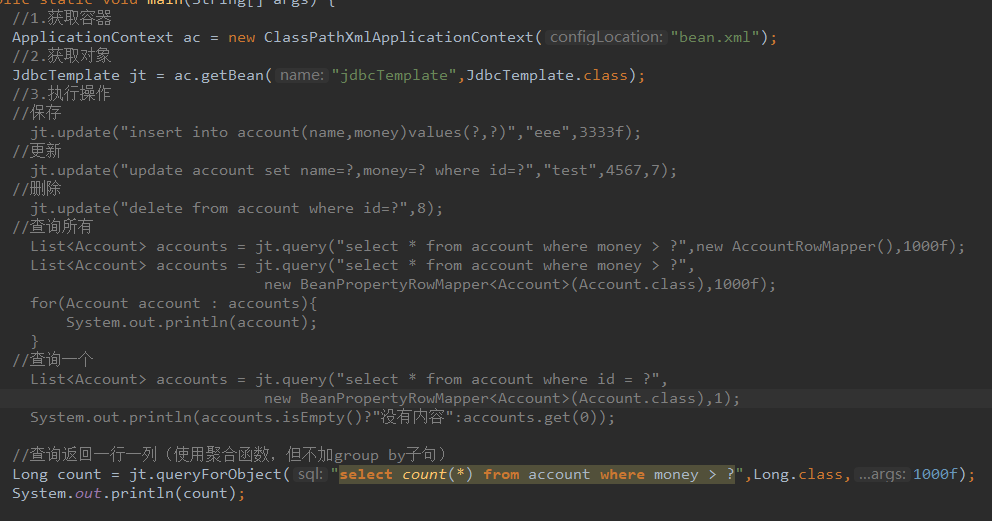
### 视频详情



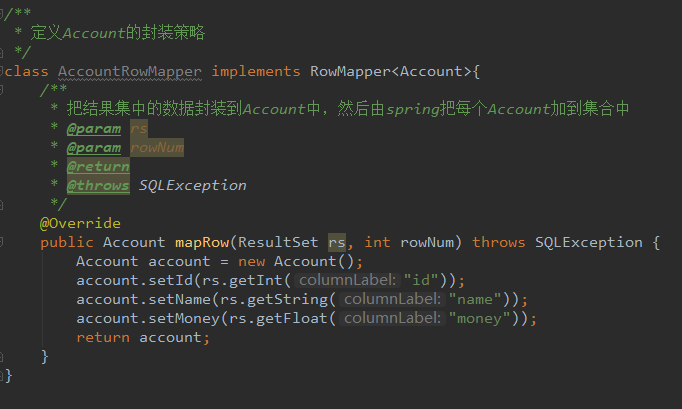
### 总结与补充

### 课堂提问与练习

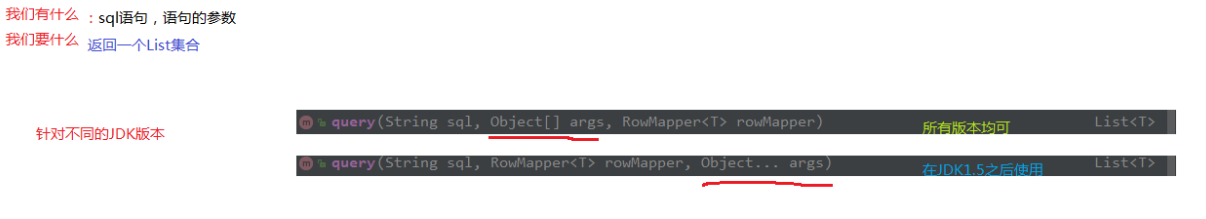
CRUD的测试类



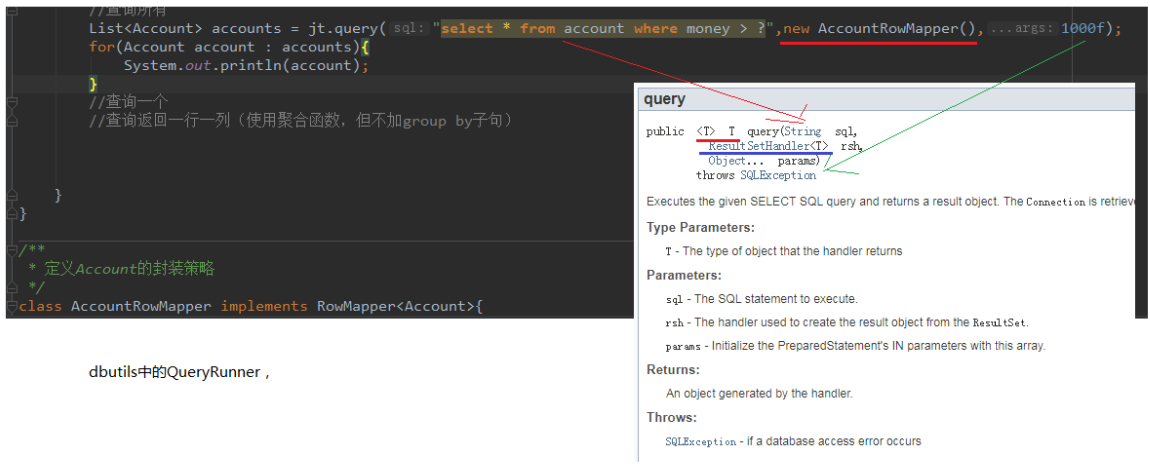
查询需要封装数据的类



查询方法分析



DBUtils与jdbcTemplate查询的对比



### 习题答案

无

## JdbcTemplate在Dao中的使用

### 知识概述

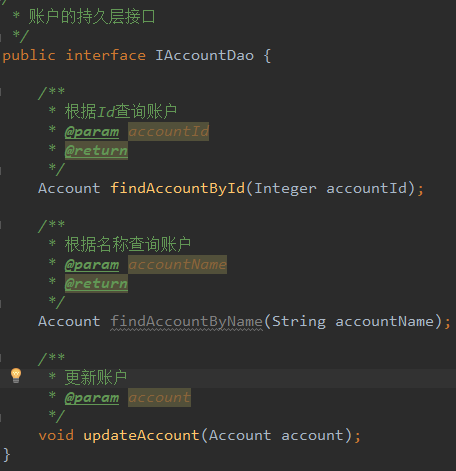
JdbcTemplate在dao层中的使用，具体代码编写

### 视频详情

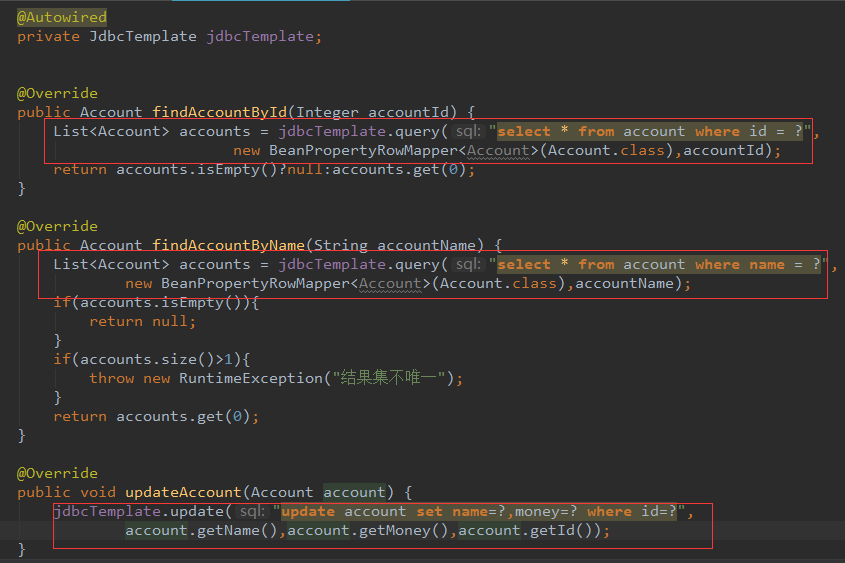


### 总结与补充

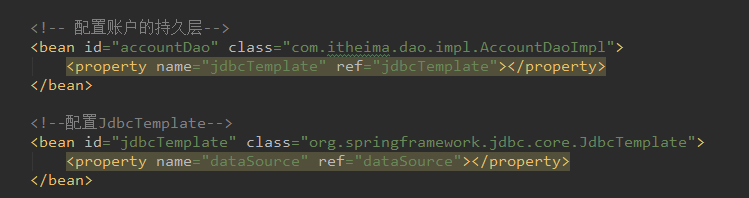
创建accoutDao接口和实现类



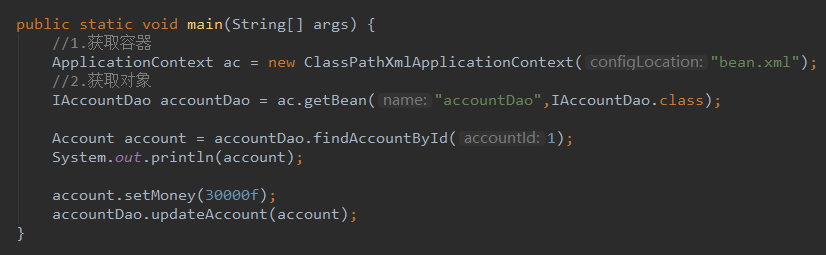
实现类



bean.xml文件中配置



测试类：



### 课堂提问与练习

无

# 第三堂课

**本节知识点**

3.1 JdbcDaoSupport的使用以及Dao的两种编写方式

3.2 作业-基于XML的AOP实现事务控制

3.3 作业-基于注解的AOP实现事务控制及问题分析

**本节目标**

能够知道更改后的Dao继承JdbcDaoSupport使用原理

能够使用aop的xml的方式实现转账的事务控制

能够使用基于注解的AOP实现事务控制

**视频时间：**

37分54秒

## JdbcDaoSupport的使用以及Dao的两种编写方式

### 知识概述

两种dao使用jdbcTemplate的方式介绍

1，原始，不继承任何类，直接spring管理

2，继承JdbcDaoSupport这个类

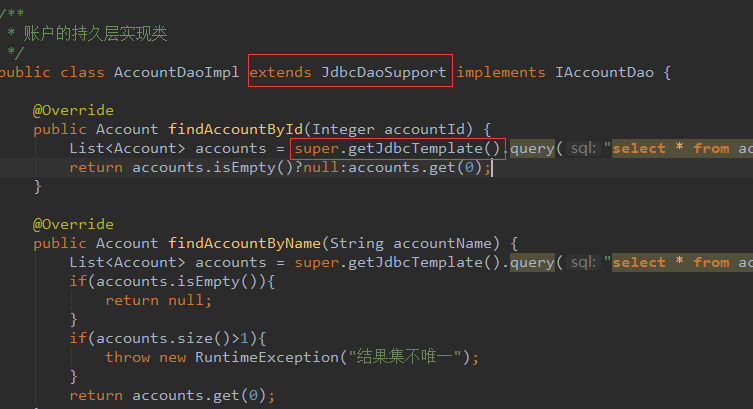
### 视频详情



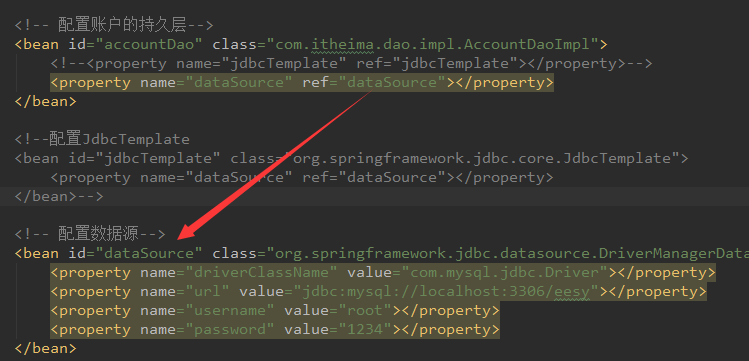
### 总结与补充

目前代码中的分析，解决重复代码dao可以继承JdbcDaoSupport

修改后的dao

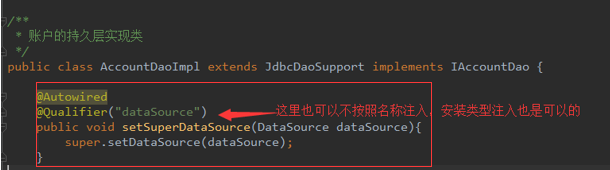


修改后的配置文件



**补充：**

如果dao类继承了JdbcDaoSupport这个类，用注解可以的方式实现注入DataSource，请看一下代码：



### 课堂提问与练习

无

### 习题答案

无

## 作业-基于XML的AOP实现事务控制

### 知识概述

修改day03中day03\_eesy\_01account转账的案例，使用aop的功能来完成事务的控制，xml文件配置的方式完成

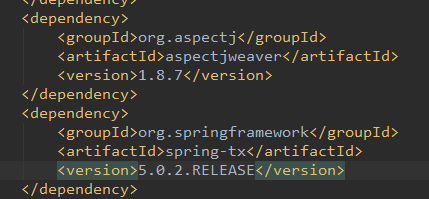
### 视频详情



### 总结与补充

新建项目day04\_eesy\_02account\_aoptx\_xml导入day03\_eesy\_01account相关代码

补充pom.文件



修改bean.xml文件

|  |
| --- |
| <!-- 配置Service -->  <bean id="accountService" class="com.itheima.service.impl.AccountServiceImpl">  <!-- 注入dao -->  <property name="accountDao" ref="accountDao"></property>  </bean>  <!--配置Dao对象-->  <bean id="accountDao" class="com.itheima.dao.impl.AccountDaoImpl">  <!-- 注入QueryRunner -->  <property name="runner" ref="runner"></property>  <!-- 注入ConnectionUtils -->  <property name="connectionUtils" ref="connectionUtils"></property>  </bean>  <!--配置QueryRunner-->  <bean id="runner" class="org.apache.commons.dbutils.QueryRunner" scope="prototype"></bean>  <!-- 配置数据源 -->  <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">  <!--连接数据库的必备信息-->  <property name="driverClass" value="com.mysql.jdbc.Driver"></property>  <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/eesy"></property>  <property name="user" value="root"></property>  <property name="password" value="1234"></property>  </bean>  <!-- 配置Connection的工具类 ConnectionUtils -->  <bean id="connectionUtils" class="com.itheima.utils.ConnectionUtils">  <!-- 注入数据源-->  <property name="dataSource" ref="dataSource"></property>  </bean>  <!-- 配置事务管理器-->  <bean id="txManager" class="com.itheima.utils.TransactionManager">  <!-- 注入ConnectionUtils -->  <property name="connectionUtils" ref="connectionUtils"></property>  </bean>  <!--配置aop-->  <aop:config>  <!--配置通用切入点表达式-->  <aop:pointcut id="pt1" expression="execution(\* com.itheima.service.impl.\*.\*(..))"></aop:pointcut>  <aop:aspect id="txAdvice" ref="txManager">  <!--配置前置通知：开启事务-->  <aop:before method="beginTransaction" pointcut-ref="pt1"></aop:before>  <!--配置后置通知：提交事务-->  <aop:after-returning method="commit" pointcut-ref="pt1"></aop:after-returning>  <!--配置异常通知：回滚事务-->  <aop:after-throwing method="rollback" pointcut-ref="pt1"></aop:after-throwing>  <!--配置最终通知：释放连接-->  <aop:after method="release" pointcut-ref="pt1"></aop:after>  </aop:aspect>  </aop:config> |

### 课堂提问与练习

无

### 习题答案

无

## 作业-基于注解的AOP实现事务控制及问题分析

### 知识概述

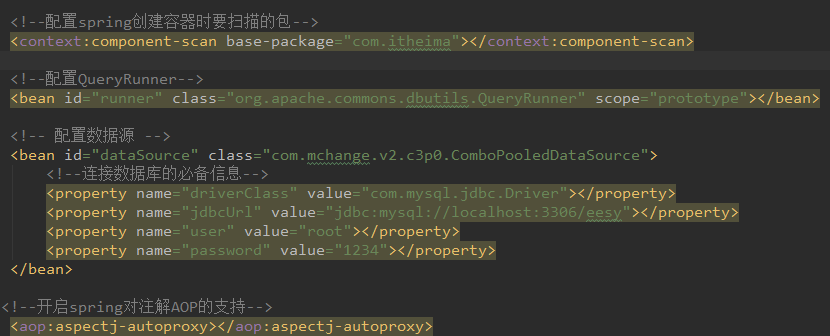
### 视频详情



### 总结与补充

修改代码：

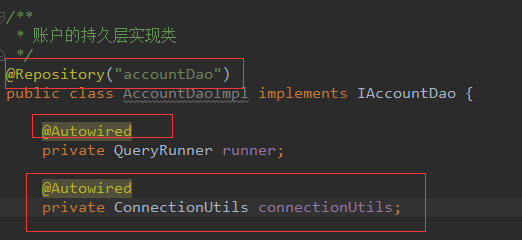
bean.xml文件



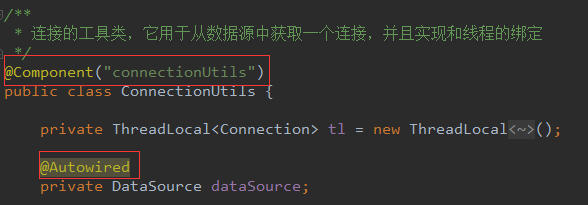
AccountServiceImpl.java



AccountDaoImpl.java



ConnectionUtils.java



TransactionManager.java





### 课堂提问与练习

无

### 习题答案

无

## 作业-基于注解的AOP实现事务控制及问题分析2

### 知识概述

使用注解的问题分析并使用环绕通知解决问题

### 视频详情



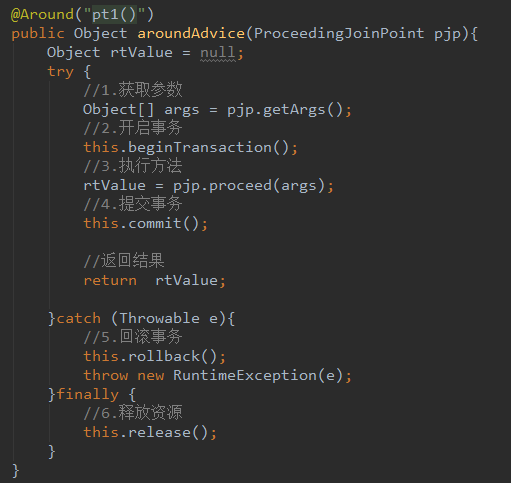
### 总结与补充

问题分析，目前使用组合通知发生的问题，

看图分析注解中的问题，图片名称：注解AOP控制事务的问题分析.png

解决方案是使用环绕通知解决

注意：去除环绕通知以外的通知



### 课堂提问与练习

无

### 习题答案

无

# 第四堂课

**本节知识点**

4.1 spring中事务控制的一组API

4.2 spring事务控制的代码准备

4.3 spring基于XML的声明式事务控制-配置步骤

**本节目标**

**视频时间：**

## spring中事务控制的一组API

### 知识概述

介绍了spring关于事务提供的三个接口

PlatformTransactionManager

TransactionDefinition

TransactionStatus

### 视频详情



### 总结与补充

**Spring 事务控制我们要明确的**

第一：JavaEE 体系进行分层开发，事务处理位于业务层，Spring 提供了分层设计 业务层的事务处理解决方案。

第二：spring 框架为我们提供了一组事务控制的接口。具体在后面的第二小节介绍。这组接口是在spring-tx-5.0.2.RELEASE.jar 中。

第三：spring 的事务控制都是基于 AOP 的，它既可以使用编程的方式实现，也可以使用配置的方式实现。我们学习的重点是使用配置的方式实现。

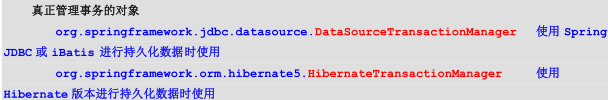
**Spring 中事务控制的 API 介绍**

1. PlatformTransactionManager

此接口是 spring 的事务管理器，它里面提供了我们常用的操作事务的方法，如下图：



我们在开发中都是使用它的实现类，如下图：



1. TransactionDefinition

它是事务的定义信息对象，里面有如下方法：



**事务的隔离级别**



**事务的传播行为**

REQUIRED:如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中。一般的选择（默认值）

SUPPORTS:支持当前事务，如果当前没有事务，就以非事务方式执行（没有事务）

MANDATORY：使用当前的事务，如果当前没有事务，就抛出异常

REQUERS\_NEW:新建事务，如果当前在事务中，把当前事务挂起。

NOT\_SUPPORTED:以非事务方式执行操作，如果当前存在事务，就把当前事务挂起

NEVER:以非事务方式运行，如果当前存在事务，抛出异常

NESTED:如果当前存在事务，则在嵌套事务内执行。如果当前没有事务，则执行 REQUIRED 类似的操作。

**超时时间**

默认值是-1，没有超时限制。如果有，以秒为单位进行设置。

**是否是只读事务**

建议查询时设置为只读。

1. TransactionStatus

此接口提供的是事务具体的运行状态，方法介绍如下图：



### 课堂提问与练习

事务的传播行为中REQUIRED是什么含义？

### 习题答案

如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中。一般的选择（默认值）

## spring事务控制的代码准备

### 知识概述

转账的代码准备

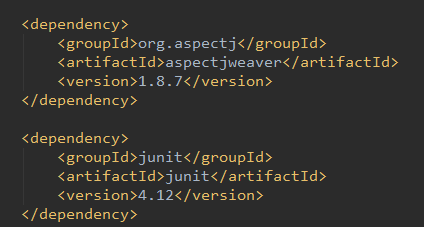
### 视频详情



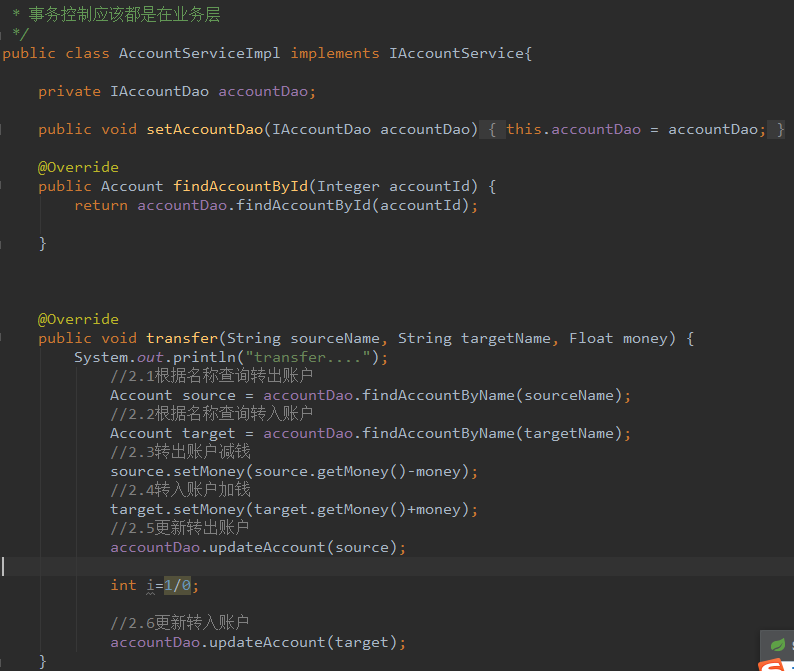
### 总结与补充

创建工程day04\_eesy\_04tx，导入关于转账的代码

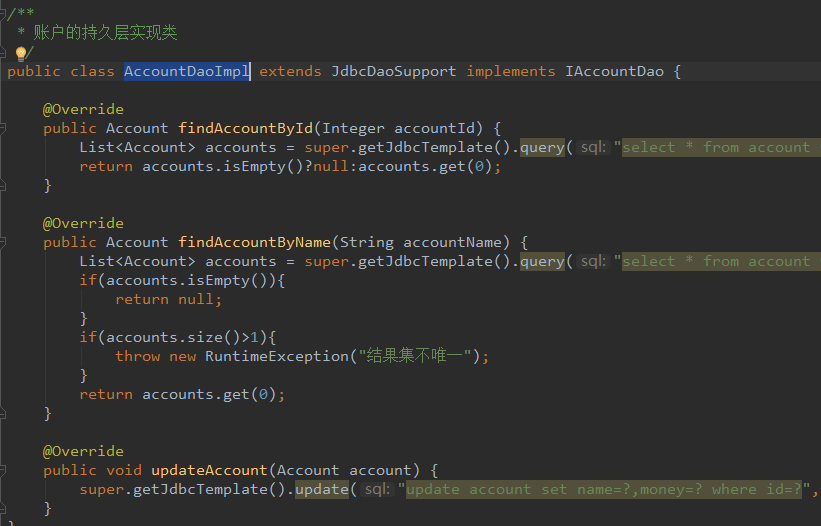
pom.xml文件中新增两个依赖



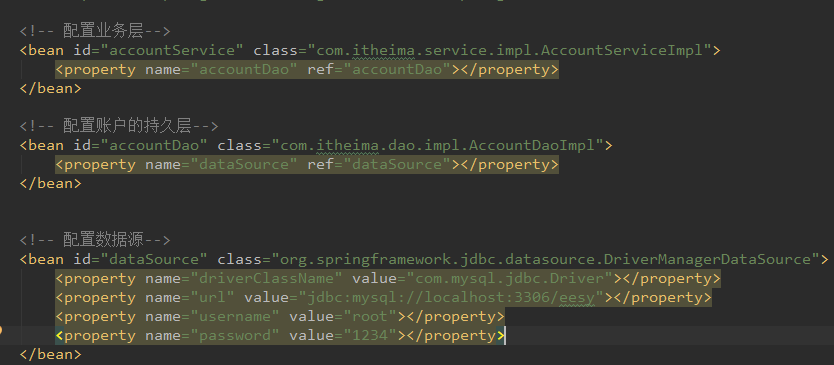
AccountServiceImpl.java



AccountDaoImpl.java



bean.xml



测试代码：



### 课堂提问与练习

无

### 习题答案

无

## spring基于XML的声明式事务控制-配置步骤

### 知识概述

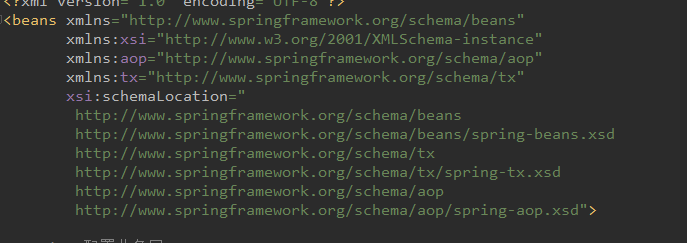
Spring中关于事务的xml的配置

### 视频详情



### 总结与补充

**导入事务相关名称空间**



**事务配置步骤**

1、配置事务管理器

2、配置事务的通知

此时我们需要导入事务的约束 tx名称空间和约束，同时也需要aop的

使用tx:advice标签配置事务通知

属性：

id：给事务通知起一个唯一标识

transaction-manager：给事务通知提供一个事务管理器引用

3、配置AOP中的通用切入点表达式

4、建立事务通知和切入点表达式的对应关系

5、配置事务的属性

是在事务的通知tx:advice标签的内部

Bean.xml文件



### 课堂提问与练习

无

### 习题答案

无

# 第五堂课

**本节知识点**

5.1 spring基于注解的声明式事务控制

5.2 spring基于纯注解的声明式事务控制

**本节目标**

**视频时间：**

## spring基于注解的声明式事务控制

### 知识概述

使用注解的方式的进行事务的控制

### 视频详情



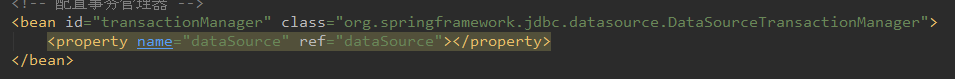
### 总结与补充

新建项目day04\_eesy\_06tx\_anno，导入上一个工程的代码

在bean.xml文件中加入context的相关名称空间

**spring中基于注解 的声明式事务控制配置步骤**

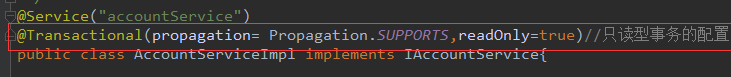
1. 配置事务管理器



1. 开启spring对注解事务的支持



3、在需要事务支持的地方使用@Transactional注解



### 课堂提问与练习

在service层用哪个注解进行事务控制？

### 习题答案

@Transactional

## spring基于纯注解的声明式事务控制

### 知识概述

修改之前的bean.xml文件，使用纯注解的方式进行管理对象

### 视频详情

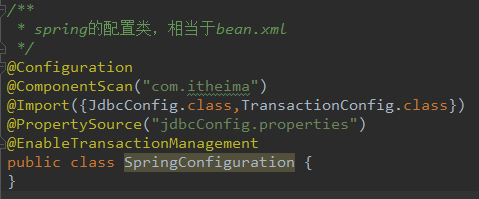


### 总结与补充

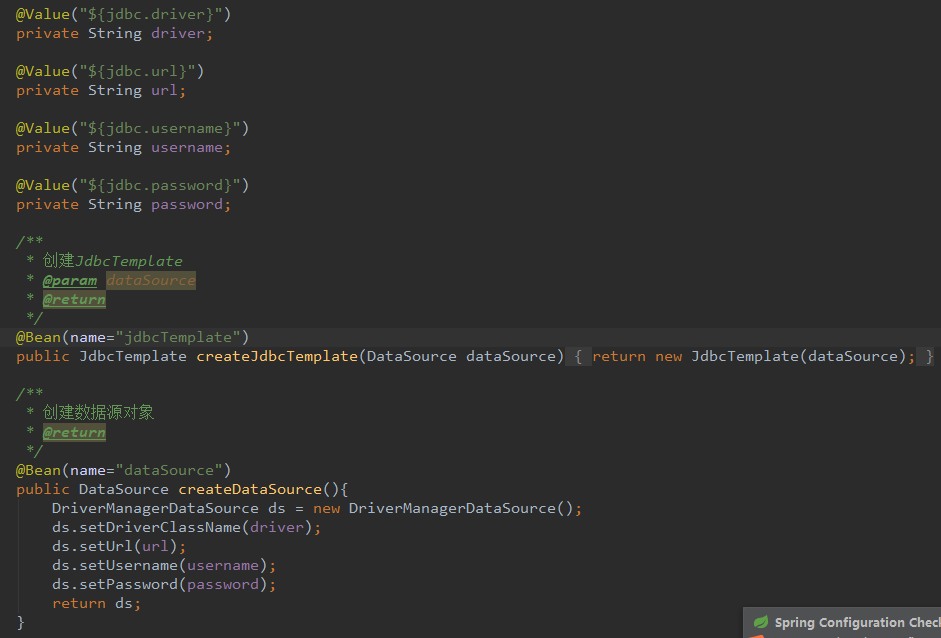
创建工程：day04\_eesy\_07anno\_tx\_withoutxml导入上一个项目中的所有代码

创建一个spring的配置类：

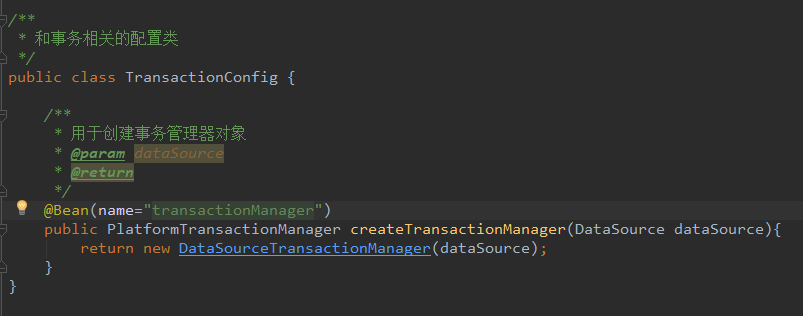
SpringConfiguration.java



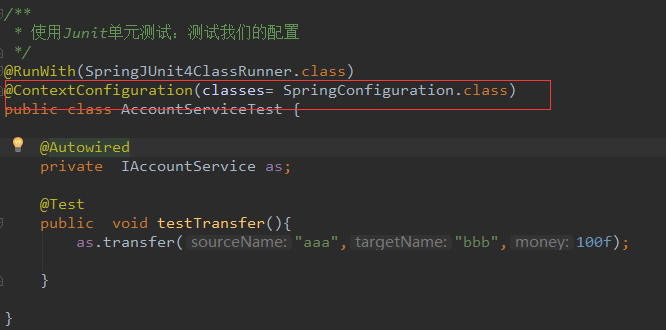
JdbcConfig.java



TransactionConfig.java



修改测试类：



### 课堂提问与练习

无

### 习题答案

无

# 第六堂课

**本节知识点**

**本节目标**

**视频时间：**

## spring编程式事务控制1-了解

### 知识概述

编程式事务的代码准备

### 视频详情

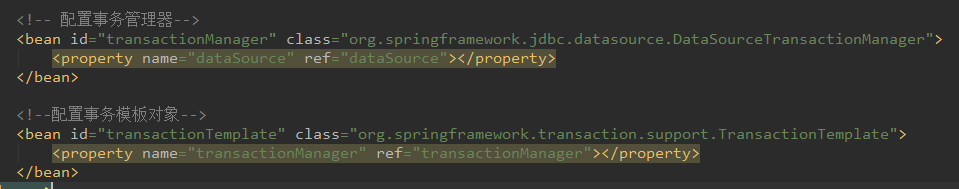


### 总结与补充

创建项目day04\_eesy\_08account\_tx导入day04\_eesy\_04tx这个工程中的所有代码

在原来的代码基础上修改bean.xml文件

配置事务管理器和事务管理模板



### 课堂提问与练习

无

### 习题答案

无

## spring编程式事务控制2-了解

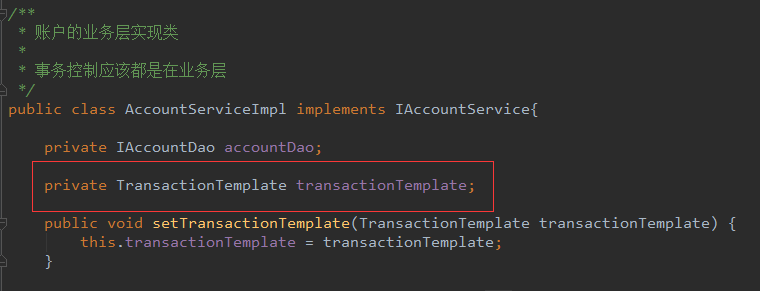
### 知识概述

### 视频详情

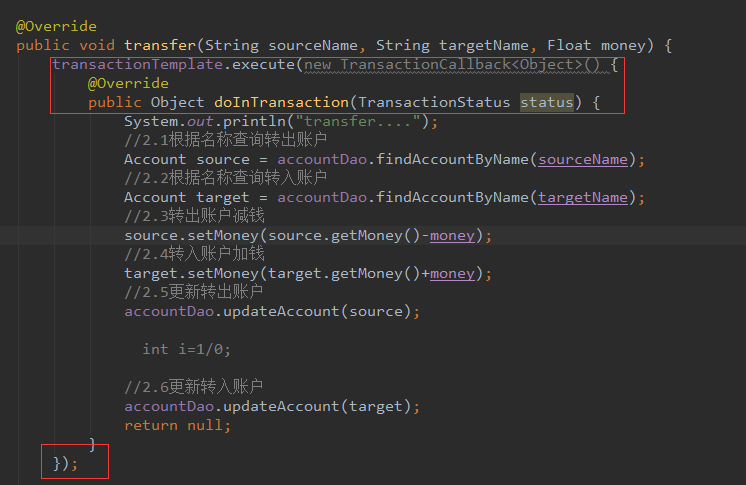


### 总结与补充

在service的实现类中新增TransactionTemplate



转账的方法：



这种方式耦合度比较高，并且产生大量的重复代码，不建议采用

### 课堂提问与练习

无

### 习题答案

无

## spring5新特性的介绍

### 知识概述

介绍一些spring5的一些新特性

### 视频详情



### 总结与补充

**jdk 版本要求**

spring5.0 在 2017 年 9 月发布了它的 GA（通用）版本。该版本是基于 jdk8 编写的，所以 jdk8 以下版本将无法使用。同时，可以兼容 jdk9 版本。tomcat 版本要求 8.5 及以上。

注：

我们使用 jdk8 构建工程，可以降版编译。但是不能使用 jdk8 以下版本构建工程。

由于 jdk 和 tomcat 版本的更新，我们的 IDE 也需要同时更新。(目前使用的 eclipse 4.7.2)

**利用 jdk8更新的内容**



**核心容器的更新**

**JetBrains Kotlin 语言支持**

**响应式编程风格**

**Junit5 支持**

**依赖类库的更新**

### 课堂提问与练习

无

### 习题答案

无