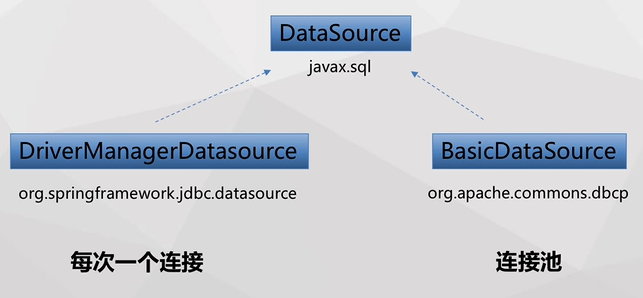
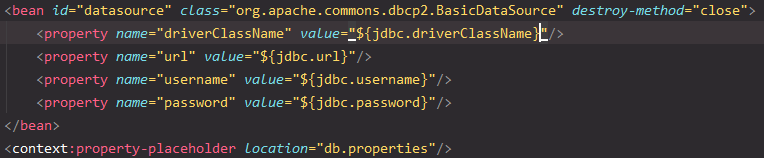
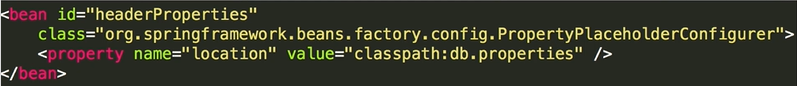
Spring JDBC

1. DAO
   1. Data Access Object
   2. 数据访问接口
2. ORM
   1. Object Relation Mapping
   2. 对象关系映射
3. Spring jdbc.jar

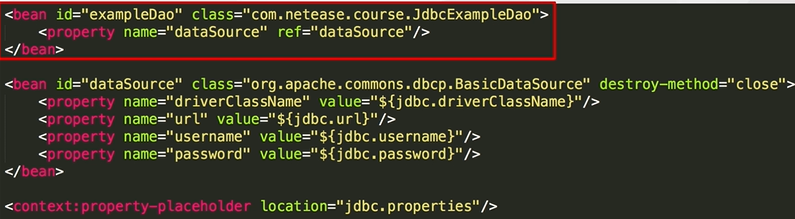
# DataSource



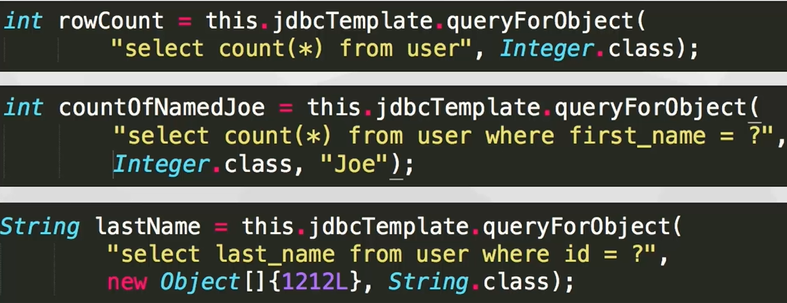
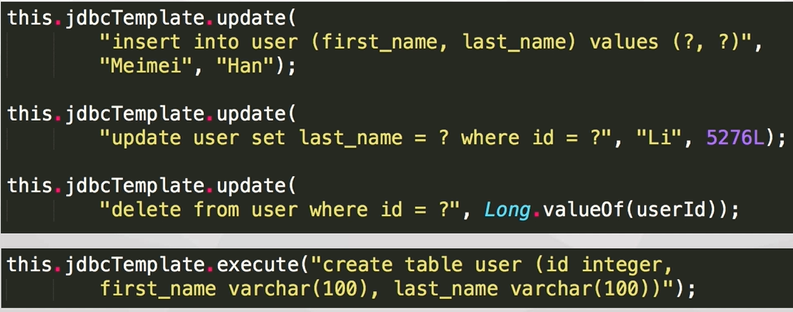
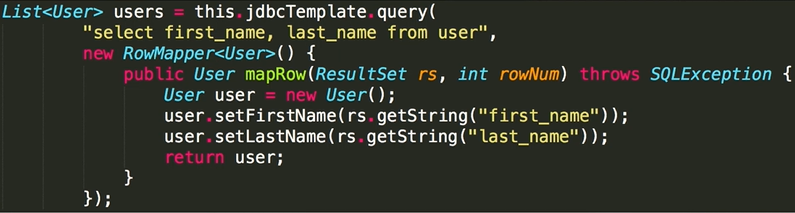
1. 配置文件
   1. 或

# JDBCTemplete

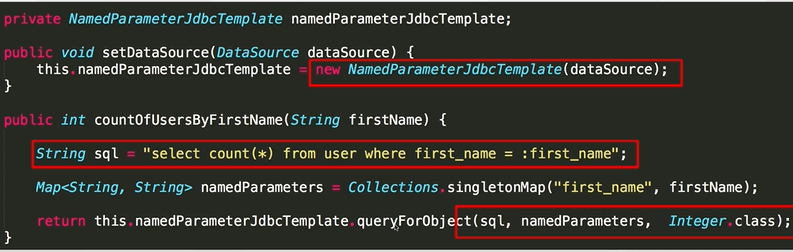
## 定义JDBCTemplate

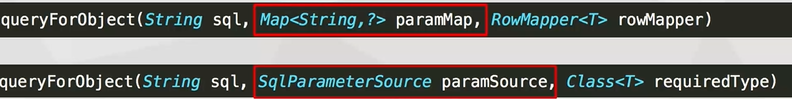
* 1. 配置——xml方式
  2. 配置——注解方式

## 操作

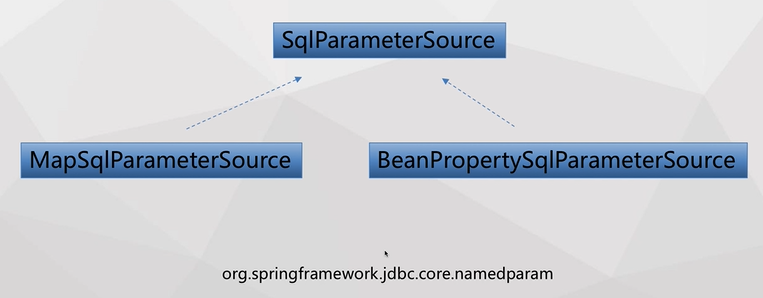
1. 查询
2. 插入
3. 返回一个对象

## NamedParameterJdbcTemplate





1. SqlParameterSource



1. BeanPropertySqlParameterSource 

## 异常处理

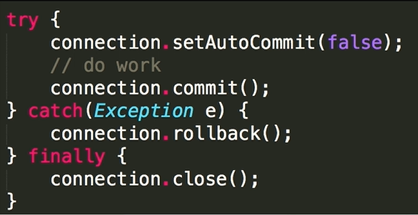


1. Spring-jdbc 可以不用try-catch

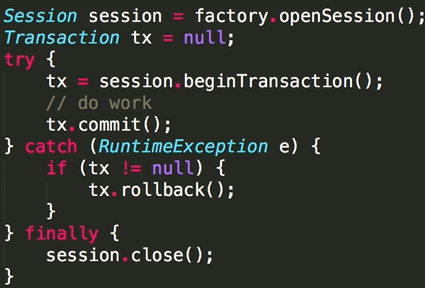
# 事务

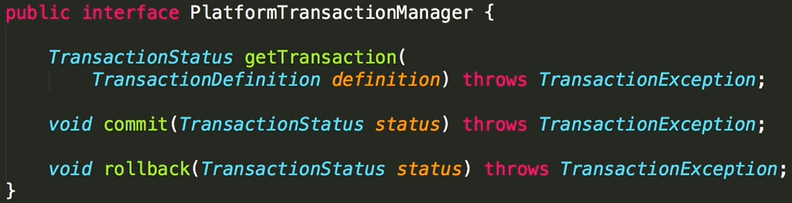
## 事务管理

1. JDBC事务

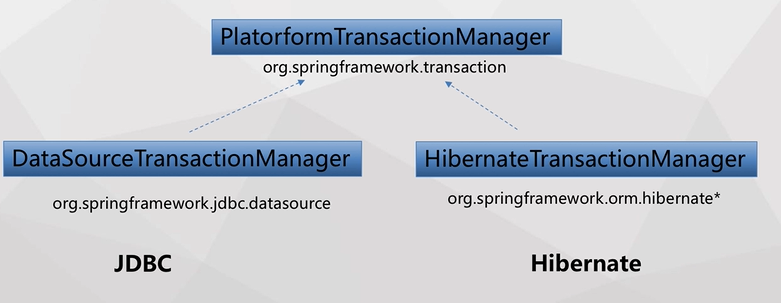


1. Hibernate事务



1. Spring事务管理
   1. 统一的事务编程模型
   2. 编程式事务及声明式事务（AOP）

## 事务管理器



## TransactionDefination

1. getName：事务名称
2. getIsolationLevel：隔离级别
3. getPropagationBehaiver：传播行为
4. getTimeout：超时时间
5. isReadOnly：是否只读事务

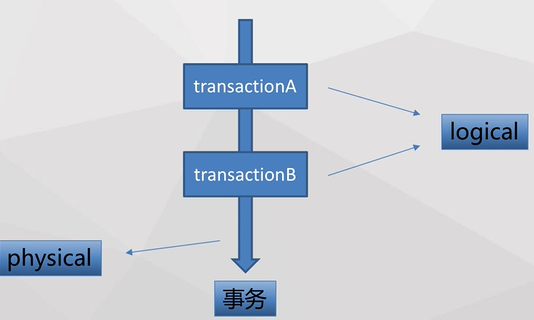
## TransactionStatus

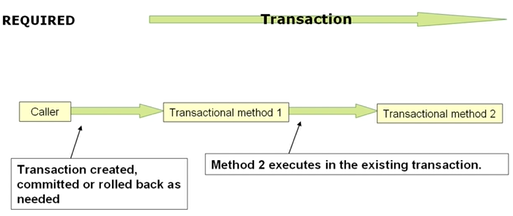
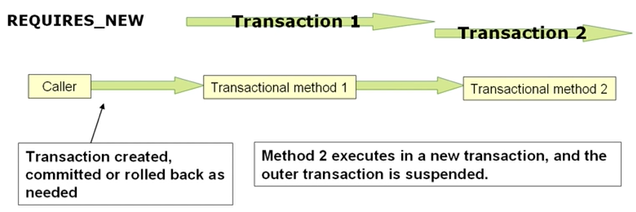
1. isNewTransaction：是否是新事务
2. hasSavepoint：是否有savePoint
3. isCompleted：是否已完成
4. isRollbackOnly：事务结果是否是rollback-only
5. setRollbackOnly：设置事务为rollback-only（TransactionTemplate）

## 隔离级别

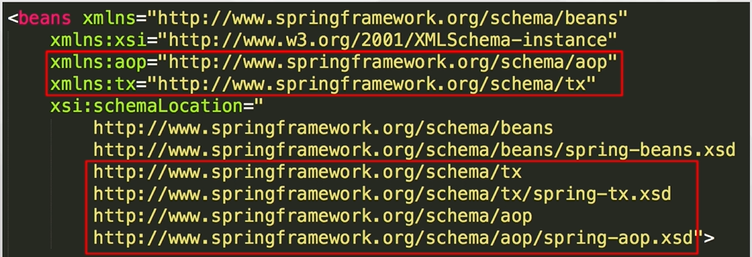
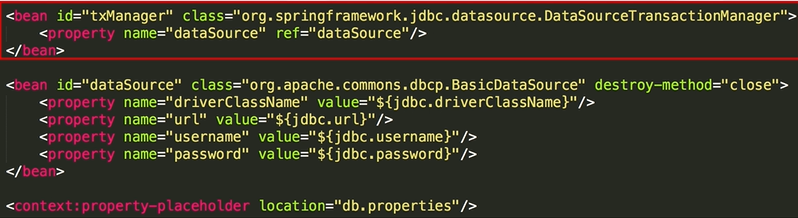
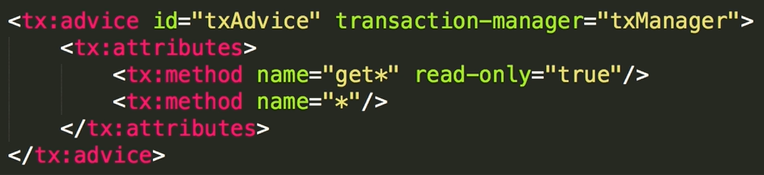
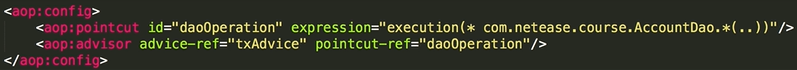
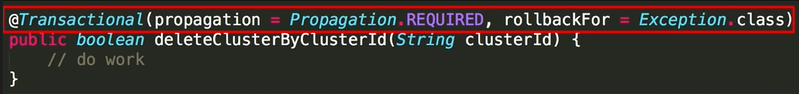
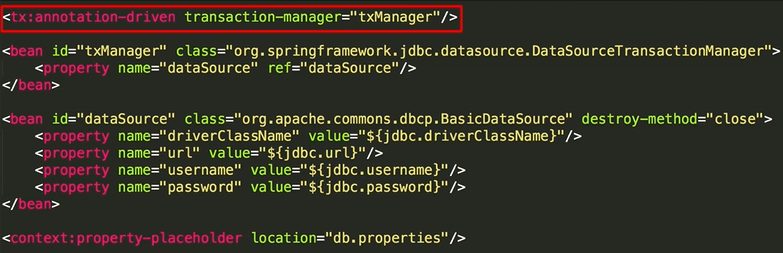
1. ISOLATION\_READ\_UNCOMMITTED：读未提交
2. ISOLATION\_READ\_COMMITTED：读提交
3. ISOLATION\_REPEATABLE\_READ：串行化
4. ISOLATION\_DEFAULT：默认

## 传播行为



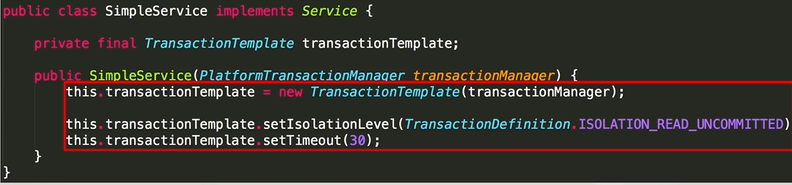
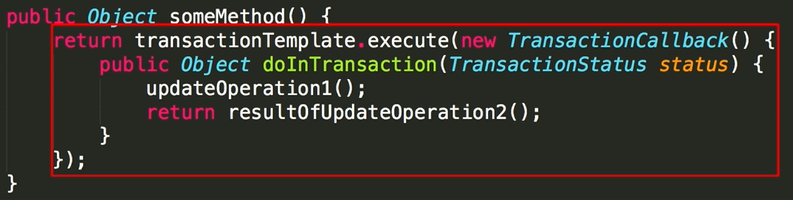
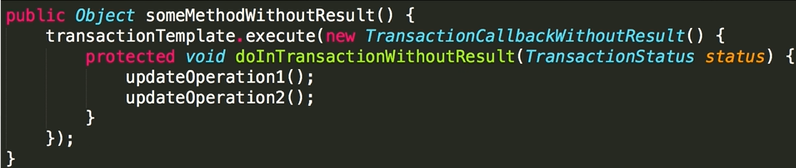
1. PROPAGATION\_MANDATORY
   1. 必须在一个事务中运行，不存在则抛出异常
2. PROPAGATION\_NEVER
   1. 不应该在事务中运行，存在则抛异常
3. PROPAGATION\_NOT\_SUPPORTED
   1. 不应该在事务中运行，存在则挂起
4. PROPAGATION\_SUPPORTS
   1. 不需要事务，有则在事务中执行
5. PROPAGATION\_REQUIRED
   1. 必须在事务中执行，如果不存在，则启动新事物
   2. 内部事务会影响外部事物
6. PROPAGATION\_NESTED
   1. 必须在事务中执行，如果不存在，则启动新事物
   2. 事务之间相互不影响（savepoint）
7. PROPAGATION\_REQUIRES\_NEW
   1. 必须在新事物中执行，挂起当前事务
   2. 独立physical事务

## 声明式事务

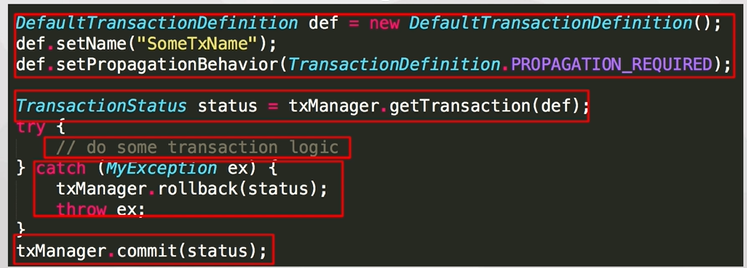
1. 添加schema
2. 定义事务管理器
3. 定义事务advice
4. 定义pointcut
5. 配置<tx:method/>
   1. name：匹配的函数名称，支持\*匹配
   2. propagation：事务传播行为
   3. isolation：事务隔离级别
   4. timeout：超时
   5. read-only：是否只读事务
   6. rollback-for：触发回滚的异常，逗号分隔
   7. no-rollback-for：不触发回滚的异常，逗号分隔
6. @Transactional
   1. Value：使用的TransactionManager
   2. Propagation：事务传播行为
   3. Isolation：事务隔离级别
   4. Timeout：超时
   5. ReadOnly：是否只读事务
   6. rollbackFor：触发回滚的异常类对象数组
   7. rollbackForClassName：触发回滚的异常类名称数组
   8. noRollbackFor：不触发回滚的异常类对象数组
   9. noRollbackForClassName：不触发回滚的异常类名称数组

## 编程式事务

### TransactionTemplate

1. 定义TranscationTemplate
2. 使用TransactionTemplate
   1. 有返回值
   2. 无返回值

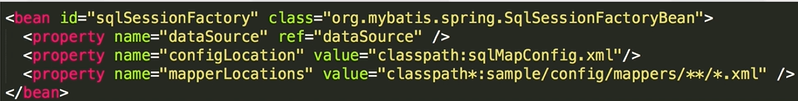
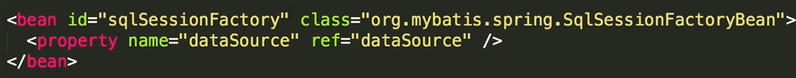
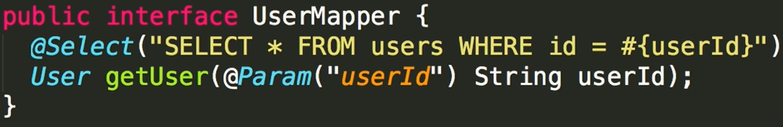
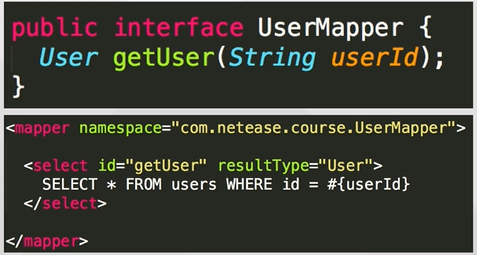
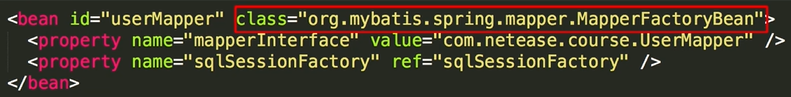
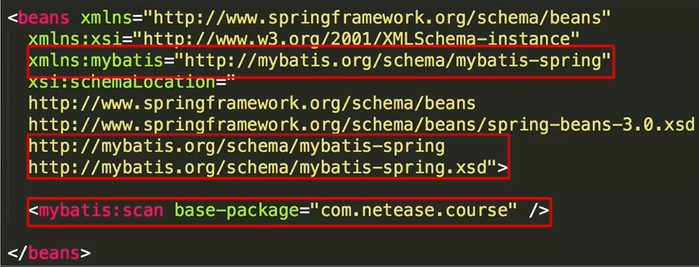
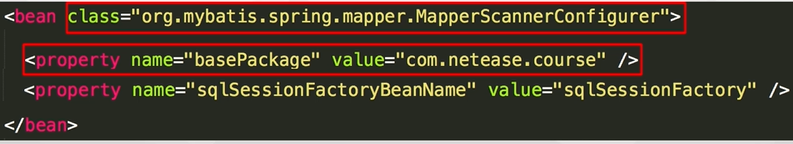
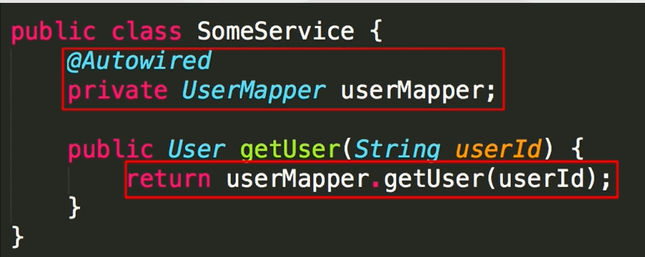
### PlatformTransactionManager的实现



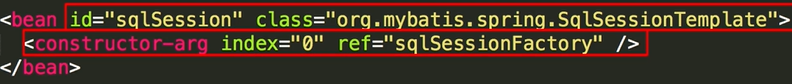
## 编程式和声明式比较

1. 编程式，事务操作少
2. 声明式，事务操作多

# Mybatis整合

1. 
2. 添加mybatis-spring依赖和mybatis依赖
3. 定义SqlSessionFactoryBean
4. Mapper
   1. 定义mapper
      1. 注解方式
      2. Xml配置文件方式
   2. 定义Mapper Bean
      1. Xml配置文件方式
      2. 自动扫描
         1. 或者
   3. 使用Mapper

## SqlSessionTemplate

1. 定义
2. 使用

## 事务管理

1. 向用户透明
2. 支持声明式事务和编程式事务

## 其他

1. Bean中的属性与数据库表中的字段名称的映射