

1、什么是事务

事务就是多条sql语句的集合，就是在代码实现某个功能时，需要用到多条DML语句（insert、update、delete）。要求这些sql语句必须同时成功，或者同时失败，他们是一个整体。

2、什么时候使用事务

当我们的操作涉及到多个表和多条sql语句对表中数据的修改操作时，希望这些sql语句只能同时成功或失败才算是完成这个功能，就需要用到事务机制，提交和回滚。如转账操作，一条sql语句对一个账户进行金额删减的修改，另一条sql语句对另一个账号进行金额的增加，两条sql语句要同时成功或失败才能确保业务的安全进行。

Java代码中编写程序完成功能，需要控制事务，是写在service类中的业务方法中的，业务方法中或多次调用dao接口中的访问数据库的方法，这就执行了多条sql语句。

3、处理事务的多种方式

使用传统JDBC的方式访问数据库处理事务时，是Connection conn，然后conn.commit()和conn.rollback()

使用mybatis的方式访问数据库处理事务时，是SqlSession sqlSession，然后sqlSession.commit()和sqlSession.rollback()

使用hibernate的方式访问数据库处理事务时，是Session session，然后session.commit()和session.rollback()

4、3中事务处理的缺点

1)、不同的数据库访问技术，处理事务使用的对象、方法都不同。这就需要我们了解不同数据库访问技术使用的事务原理

2)、还需要掌握多种事务处理的逻辑。什么时候提交事务，什么时候回滚事务

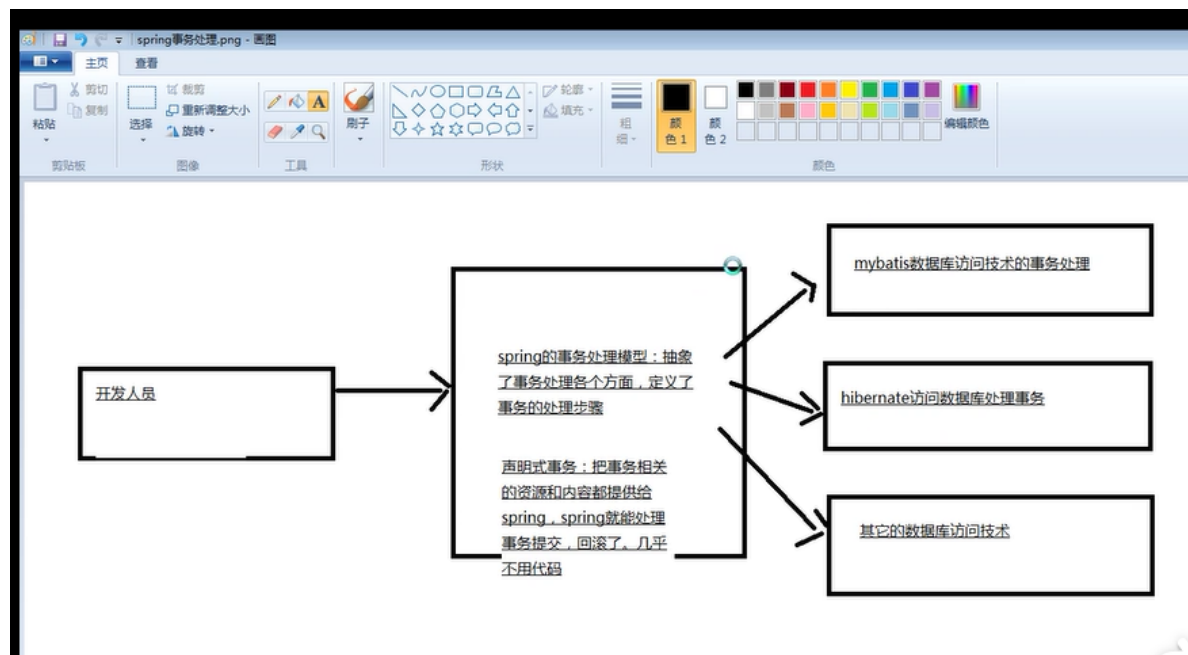
3)、处理事务的方法过多

总之，多种数据库访问技术过多，致使处理事务的机制、对象和方法要掌握的就多。

5、解决方式

Spring框架提供了一种处理事务的统一模型，将处理事务的步骤统一起来，通过提供不同数据库访问技术的信息让Spring判断生成什么样的事务处理对象。

使用Spring事务统一机制，能够完成各种数据库访问方式的事务处理。



6、使用Spring事务机制需要怎么做

Spring处理事务的模型，使用的步骤都是固定的，我们只需要把事务使用的信息提供给Spring就可以了

1)、Spring事务内部提交和回滚事务，使用的是事务管理对象，代替我们完成commit和rollback。事务管理器是一个接口和该接口的众多实现类。

接口：PlatformTransactionManager，定义了事务的一些重要方法commit、rollback等

实现类：Spring已经把每一种访问数据库的方式对应的事务处理类都创建好了，都是上面接口的实现类。

mybatis方式访问数据库的事务处理使用的是——DataSourceTransactionManager

hibernate方式访问数据库的事务处理使用的是——HibernateTransactionManager

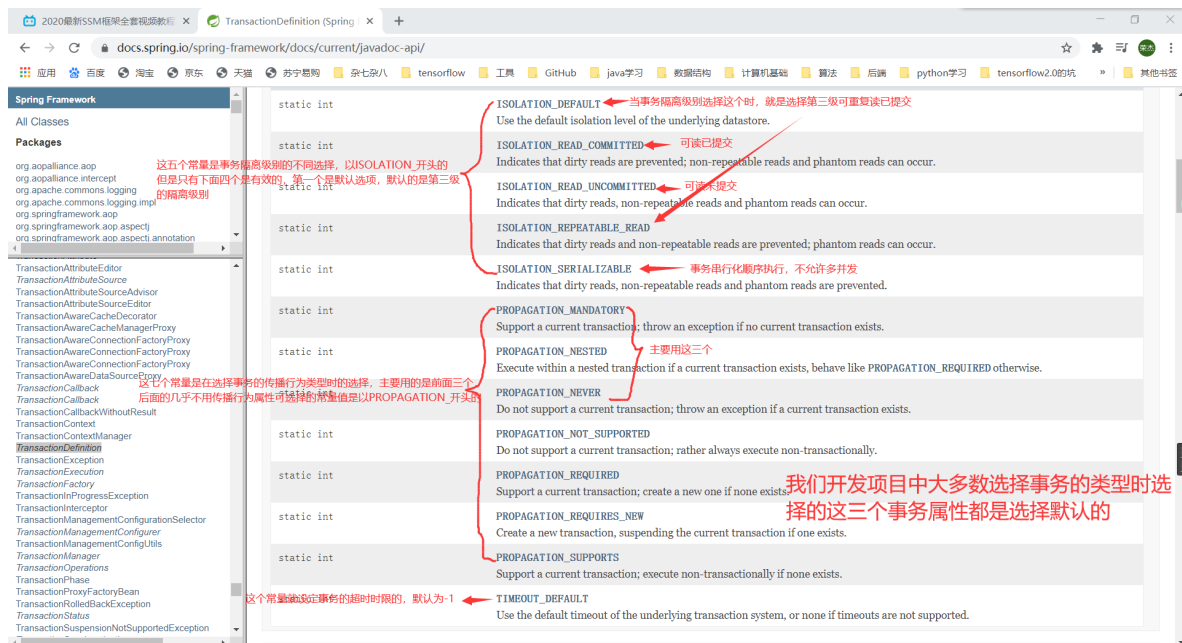
使用方式：需要告诉Spring我们使用的是哪种访问数据库的方式就行了

方法：在Spring的配置文件中使用标签声明数据库访问技术对应的事务管理器实现类。如我们使用的是mybatis，如下：

```
<bean id="xxx" class="...DataSourceTransactionManager"/>
```

2)、业务方法所需要的是什么类型的事务，包括如下：

spring中有一个接口TransactionDefinition，其中定义了描述这个事务是什么类型的三类常量：事务的隔离级别、事务的传播行为、事务默认超时时限，以及对这些的操作。我们要给一个业务方法添加一个事务，我们就需要对这三个进行设置，选择什么样的隔离级别，什么样的传播行为，多久会超时。，访问Spring官网查看帮助文档，找到该接口，查看接口中定义的属性如下图：



1、事务的隔离级别：总共有四级

spring框架中使用常量来表示这个事务的隔离级别，在TransactionDefinition接口中关于该类型的属性有如下五个：

static int ISOLATION_DEFAULT //选择这个使用默认的隔离级别，MySQL默认是第三级，Oracle是第二级

static int ISOLATION_READ_COMMITTED //第一级，读未提交。未解决任何并发问题

static int ISOLATION_READ_UNCOMMITTED //第二级，读已提交。解决脏读，存在不可重复读和幻读

static int ISOLATION_REPEATABLE_READ //第三级，可重复读。解决脏读、不可重复读，存在幻读

static int ISOLATION_SERIALIZABLE //第五级，串行化。不解决任何并发问题

2、事务的超时时间。表示一个方法的最长执行时间。如果方法执行超过设定的，事务就会回滚。

单位是秒，整数值，默认是-1。因为一个业务方执行影响因素太多，不确定的也多，所以一般使用默认的，不设置超时时

间

3、事务传播行为：有七个，控制业务方法是不是有事务的，是什么样的事务。

表示在业务方法调用时，事务在方法之间是如何使用的。

//主要用到的是下面三个：

//指定方法必须在一个指定的事务内执行，若当前存在事务，把当前事务作为指定事务；若没有，创建一个新建的指定事务，让方法在这个业务中执行。这种传播行为是最常见的，也是spring默认使用的

static int PROPAGATION_REQUIRED

//选择下面的常量值，指定当前方法有事务可以执行，没有也可以非事务方式执行，如查询操作

static int PROPAGATION_SUPPORTS

//选择下面这个常量值，总会新建一个事务，当前方法有事务，就把事务先挂起，执行完新建的，在执行原来的

static int PROPAGATION_REQUIRES_NEW

//下面三个不用

static int PROPAGATION_NOT_SUPPORTED

static int PROPAGATION_MANDATORY

static int PROPAGATION_NESTED

static int PROPAGATION_NEVER

3)、Spring提交事务、回滚事务的时机

- 1、当业务方法执行成功，没有抛出异常，最后Spring在方法执行完毕后提交事务
- 2、当业务方法执行时抛出运行时异常或Error时（继承RuntimeException的异常），Spring调用事务管理器的rollback方法进行回滚操作。执行时异常包括如空指针异常、NumberFormatException等
- 3、当业务方法抛出编译时异常或Error时，默认会提交事务。

总结Spring事务：

- 1、管理事务的是：事务管理器和它的实现类
- 2、Spring的事务是一个统一的模型
 - 1)、指定要使用的事务管理器实现类，使用
 - 2)、指定哪些类哪些方法需要加入事务功能
 - 3)、指定方法需要的隔离级别、传播行为、超时