

事务的ACID

2021年10月6日 17:07

在Spring框架中的事务是比数据库中的事务更大的一个概念

数据库事务：操作数据库的一个操作序列，要么全部执行，要么全部不执行

Spring框架中的事务**不局限于对数据库的操作**，只要其中有数据源是支持事务的，它就可以把多个数据源的操作作为一个整体放到一个事务中，其中任一失败，就会回滚

事物的ACID特性

原子性 Atomicity

事务中所有操作是不可分割的，要么全部执行，要么全部不执行

一致性 Consistency

事务必须从一个一致状态变换到另一个一致性状态

当有多个事务并行执行时，执行结果一定是一致的，一致的结果和它的某一个顺序执行结果是相同的，不会说因为多个事务的执行造成最后执行的结果不同

（举例解释：假设你汇款，你的账户钱减少的同时对方账户钱会增加，即数据库中的数据总量是不变的）

隔离性 Isolation

在并行执行的事务，是不受其他事务干扰的

每一个事务在执行过程中所看到的東西应该是属于它自己的（其他事务的执行对于这个事务来说是透明的）

持久性 Durability

如果事务已经结束了，提交了数据，系统一定要保证事务提交的数据会被写入，即它的修改是不可丢失的
即使事务结束后系统发生故障，这个数据也不能丢失

Spring中的事务

Spring中的事务是通过**@Transaction**来标识，可以标在Bean对象方法前，也可以标在Bean对象的类前
标在Bean对象类前会让类的所有方法都有事务，在方法前标识这个方法是有事务的

我们经常在**Service层的方法**前标上这个注解，因为Controller层的对象主要是负责数据的合法性检查，以及返回的数据的封装。合法性检查在所有操作之前，封装不容易出错，所以它们不涉及事务回滚

@Transaction的属性

- propagation
用来标注事务的传播方式
- isolation
用来标注事务的隔离级别
- readOnly
用来标识当前事务是一个只读事务
- rollbackFor
- rollbackForClassName
这两个表示会引起事务异常的东西，一个用名字来标识，一个用对象

- noRollbackFor
- noRollbackForClassName

这两个表示不会引起事务异常的东西，一个用名字来标识，一个用对象

propagation属性

1. REQUIRED (默认)

如果当前没有事务，它就会开启一个新的事务

如果当前有事务，它就会沿用这个事务

(为什么当前可能会有事务：

因为我们会把注解写在服务层的方法前，这个方法可能会被另一个服务层的方法调用，如果那个方法也开启了事务，那个事务就会沿用到当前的方法来)

2. SUPPORTS

表示当前的代码是**支持事务的**

如果调用它的方法是有事务的，它就会加入这个事务，没有则不加入

3. MANDATORY

表示这个方法**必须有事务**

如果调用它的方法有事务，它就会加入这个事务，没有则抛出异常

4. REQUIRES_NEW

表示这个方法**必须新建一个事务**

不管调用它的方法是否有事务，它都会自己新建一个事务

如果调用它的方法有事务，它会把这个事务挂起，然后自己新建一个事务

5. NOT_SUPPORTED

表示当前方法**不支持事务**

如果调用它的方法没有事务，正常执行

如果调用它的方法有事务，它仍以无事务的方式执行，**会把这个事务挂起**，以无事务的方式运行完这段代码。

6. NEVER

表示当前方法**不能有事务**

如果调用它的方法没有事务，正常执行

如果调用它的方法有事务，则抛出异常

7. NESTED

表示当前方法可以**嵌套事务**

如果调用它的方法有事务，则嵌套在当前事务中

如果调用它的方法没有事务，则新建一个事务

内部事务回滚不会影响外部事务，但外部事务回滚会把内部事务一起回滚回去