# spring事务管理

事务管理是企业级应用程序开发中必不可少的技术, 用来确保数据的完整性和一致性.

事务的四个关键属性(ACID)

原子性(atomicity): 事务是一个原子操作, 由一系列动作组成. 事务的原子性确保动作 要么全部完成要么完全不起作用.

一致性(consistency): 一旦所有事务动作完成, 事务就被提交. 数据和资源就处于一种 满足业务规则的一致性状态中.

隔离性(isolation): 可能有许多事务会同时处理相同的数据, 因此每个事物都应该与其 他事务隔离开来, 防止数据损坏.

持久性(durability): 一旦事务完成, 无论发生什么系统错误, 它的结果都不应该受到影 响. 通常情况下, 事务的结果被写到持久化存储器中.

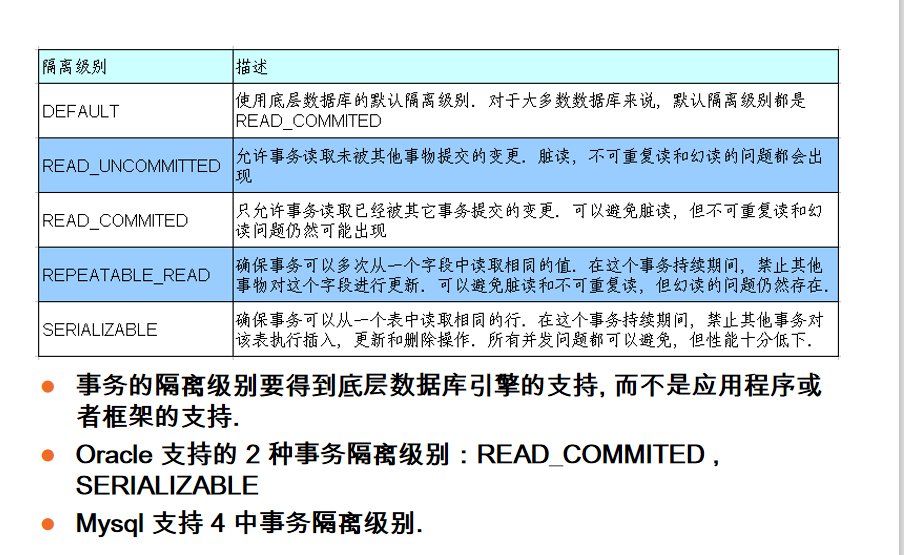
并发事务所导致的问题可以分为下面三种类型:

脏读: 对于两个事物 T1, T2, T1 读取了已经被 T2 更新但 还没有被提交的字段. 之后, 若 T2 回滚, T1读取的内容就是临时且无效的.

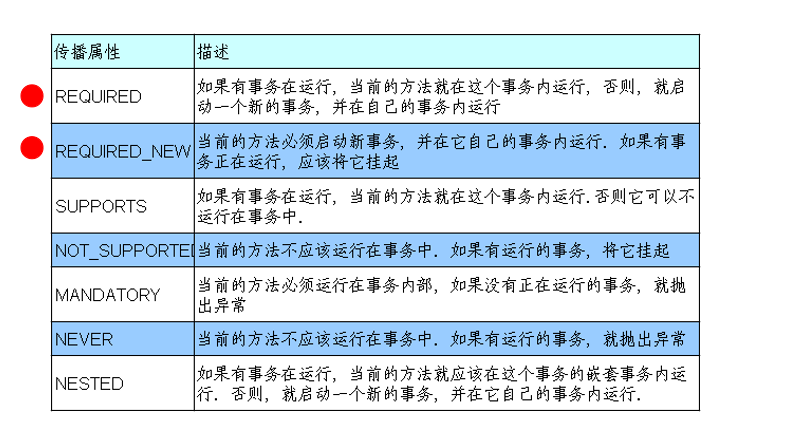
不可重复读:对于两个事物 T1, T2, T1 读取了一个字段, 然后 T2 更新了该字段. 之后, T1再次读取同一个字段, 值就不同了. update

幻读:对于两个事物 T1, T2, T1 从一个表中读取了一个字段, 然后 T2 在该表中插入了 一些新的行. 之后, 如果 T1 再次读取同一个表, 就会多出几行.insert delete

隔离级别:



事务的传播属性:



回滚属性:

默认情况下只有未检查异常(RuntimeException和Error类型的异常)会导致事务回滚. 而 受检查异常不会.

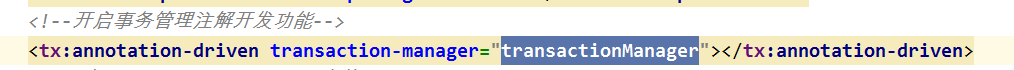
事务的回滚规则可以通过 @Transactional 注解的 rollbackFor 和 noRollbackFor 属性 来定义. 这两个属性被声明为 Class[] 类型的, 因此可以为这两个属性指定多个异 常类.

rollbackFor: 遇到时必须进行回滚

noRollbackFor: 一组异常类，遇到时必须不回滚

给予注解的spring事务管理添加步骤:

1. 添加事务管理相关的jar包
2. 配置切面对象
3. 开启事务管理注解开发功能,并且指定使用哪个事务管理器



1. 在需要添加事务的方法上添加事务注解

只读属性:readOnly

超时属性:timeout