1 Spring 事务管理器

### 1.1 事务特性

原子性：

隔离性：多个并发操作相隔离（锁）

一致性：数据一致性

持久性：

### 1.2 Spring 事务管理器用到的三个接口

PlatformTransactionManager(事务管理器)

TransactionDefinition（事务定义信息：隔离、传播、超时、只读）

TransactionStatus（事务具体运行状态）

TransactionDefinition

### 1.3 事务的隔离级别：

|  |  |
| --- | --- |
| 隔离级别 | 含义 |
| Default | 采用底层数据库默认的级别；mysql用的是REPEATABLE\_READ、oracle 用的是READ\_COMMITED |
| READ\_UNCOMMITED | 允许读取 还未提交的改变了的数据，可能出现脏读，幻读，不可重复读 |
| READ\_COMMITED | 允许在并发事务已提交 后读取 ，可防止脏读 |
| REPEATABLE\_READ | 相同字段在多次读取 数据是一致的，除非被事务本身操作改变，可防止脏读，不可重读 |
| SERIALIZABLE | 完全服从ACID,事务排队执行，不可能出现并发。 |

### 1.4 事务的传播行为

持久层

业务逻辑层

Web 层

Serivce1{

Method1{DAO.XX(); DAO2.XX();}

} DAO{}

Service2{ DAO.XX(); DAO2.XX();

} DAO2{}

问 题：如果web请你同时调用 了service1和service2，那么使用的是哪一个service中的事务？？

解决：事务的传播（解决业务层相互调用的时候事务是如何传递的）



### 1.5 Spring 事务管理方式

——编程式的

实际应用中很少用到

通过TransactionTemplated 手动管理事务

——使用XML声明配置式

代码 侵入性最小

通过AOP实现