

# Spring 声明式事务，propagation属性列表

作者: [justdoit](#)

原文链接: <https://ld246.com/article/1465264190038>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

**TransactionDefinition** 接口中定义，共有 种选项可用：

**PROPAGATION\_REQUIRED**：支持当前事务，如果当前没有事务，就新建一个事务。这是最常见的选择。

**PROPAGATION\_SUPPORTS**：支持当前事务，如果当前没有事务，就以非事务方式执行。

**PROPAGATION\_MANDATORY**：支持当前事务，如果当前没有事务，就抛出异常。

**PROPAGATION\_REQUIRES\_NEW**：新建事务，如果当前存在事务，把当前事务挂起。

**PROPAGATION\_NOT\_SUPPORTED**：以非事务方式执行操作，如果当前存在事务，就把当前事务挂起。

**PROPAGATION\_NEVER**：以非事务方式执行，如果当前存在事务则抛出异常。

**PROPAGATION\_NESTED**：支持当前事务，如果当前事务存在，则执行一个嵌套事务（还不是太明白），如果当前没有事务，就新建一个事务。

**TransactionDefinition** 中定义的 隔离（isolation）级别，有 种：

1 ISOLATION\_DEFAULT 默认的隔离级别

下面几个都是 JDBC isolation levels 对应：

2 ISOLATION\_READ\_UNCOMMITTED Connection.TRANSACTION\_READ\_UNCOMMITTED 指示防止发生脏读的常量；不可重复读和虚读有可能发生。

3 ISOLATION\_READ\_COMMITTED Connection.TRANSACTION\_READ\_COMMITTED 指示可以生脏读（dirty read）、不可重复读和虚读（phantom read）的常量。

4 ISOLATION\_REPEATABLE\_READ Connection.TRANSACTION\_REPEATABLE\_READ 指示防止发脏读和不可重复读的常量；虚读有可能发生。

5 ISOLATION\_SERIALIZABLE Connection.TRANSACTION\_SERIALIZABLE 指示防止发生脏读、不可重复读和虚读的常量。

**注**：脏读（dirty reads）：当事务读取还未被提交的数据时，就会发生这种事件。举例来说：Transaction 1 修改了行数据，然后 Transaction 2 在 Transaction 1 还未提交修改操作之前取了被修改的行。如果 Transaction 1 回滚了修改操作，那么 Transaction 2 读的数据就可以看作是从未存在过的。

**不可重复的读（non repeatable reads）**：当事务两次读取同一行数据，但每次得到的数据都不一样时，就会发这种事件。举例来说：Transaction 1 读取一行数据，然后 Transaction 2 修改删除该行并提交修改操作。当 Transaction 1 试图重新读取该行时，它就会得到不同的据值（如果该行被更新）或发现该行不再存在（如果该行被删除）。

**虚读（phantom read）**：如果符合搜索条件的一行数据在后面的读取操作中出，但该行数据却不属于最初的数据，就会发生这种事件。举例来说：Transaction 1 读取满足种搜索条件的一些行，然后 Transaction 2 插入了符合 Transaction 1 搜索条件的一个新行。如果 Transaction 1 重新执行产生原来那些行的查询，就会得到同的行。