

Spring中事务传播行为种类Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022_Spring_E4_B8_AD_E4_c104_644915.htm Spring在TransactionDefinition接口中规定了7种类型的事务传播行为，它们规定了事务方法和事务方法发生嵌套调用时事务如何进行传播：

PROPAGATION_REQUIRED 如果当前没有事务，就新建一个事务，如果已经存在一个事务中，加入到这个事务中。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 使用当前的事务，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** 如果当前存在事务，则在嵌套事务内执行。如果当前没有事务，则执行与**PROPAGATION_REQUIRED**类似的操作。在使用过程中

需要注意以下几点：1) **NESTED**和**NEW**的区别

PROPAGATION_REQUIRES_NEW 和

PROPAGATION_NESTED是容易混淆的两个传播行为

。 **PROPAGATION_REQUIRES_NEW** 启动一个新的、和外层事务无关的“内部”事务。该事务拥有自己的独立隔离级别和锁，不依赖于外部事务，独立地提交和回滚。当内部事务开始执行时，外部事务将被挂起，内务事务结束时，外部事

务才继续执行。由此可见，

PROPAGATION_REQUIRES_NEW 和

PROPAGATION_NESTED 的最大区别在于

：PROPAGATION_REQUIRES_NEW 将创建一个全新的事务，它和外层事务没有任何关系，而 PROPAGATION_NESTED 将创建一个依赖于外层事务的子事务，当外层事务提交或回滚时，子事务也会连带提交和回滚。嵌套事务不能够提交，它必须通过外层事务来完成提交的动作，外层事务的回滚也会造成内部事务的回滚。 2) 当方法被设置

为PROPAGATION_NOT_SUPPORTED时，外层业务方法的事务会被挂起，当内部方法运行完成后，外层方法的事务重新运行。如果外层方法没有事务，直接运行，不需要做任何其它的事。 3) 当业务方法被设置为PROPAGATION_NEVER

时，它将不能被拥有事务的其它业务方法调用。 4) 当业务方法被设置为PROPAGATION_MANDATORY时，它就不能被

非事务的业务方法调用。 编辑特别推荐: Java核心API需要掌握的程度 JAVA远程访问共享目录 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com