

当前位置：HollisChuang's Blog (https://www.hollischuang.com) > 框架 (https://www.hollischuang.com/archives/category/%e6%a1%86%e6%9e%b6) > spring (https://www.hollischuang.com/archives/category/%e6%a1%86%e6%9e%b6/spring) > 正文

Spring的事务管理机制 (https://www.hollischuang.com/archives/1489)

2016-05-07

分类：spring (https://www.hollischuang.com/archives/category/%e6%a1%86%e6%9e%b6/spring)

阅读(9179)

评论(1)

阿里大牛珍藏架构资料，点击链接免费获取 (https://mp.weixin.qq.com/s?biz=MzU4MDM4MzU5Mg==&mid=2247483700&idx=1&sn=444d8c096d5a4b959c5988dbd90bfbb9&chksm=fd56e733ca216e25021f14b23a6ee9f9764b8a3e115837ceaa2f3a65408674d00cbed42c78f&token=494746963&lang=zh_CN#rd).

在我的博客（http://www.hollischuang.com/）中，多篇文章介绍了事务相关的内容 (http://www.hollischuang.com/archives/tag/%E4%BA%8B%E5%8A%A1)，其中包括数据库的事务 (http://www.hollischuang.com/archives/898)的相关介绍、分布式事务 (http://www.hollischuang.com/archives/681)的相关介绍以及在Spring中使用注解进行事务的配置方式 (http://www.hollischuang.com/archives/297)等。

本文在以上文章的基础上，一起来学习一下Spring对事务的支持——Spring的事务管理机制。

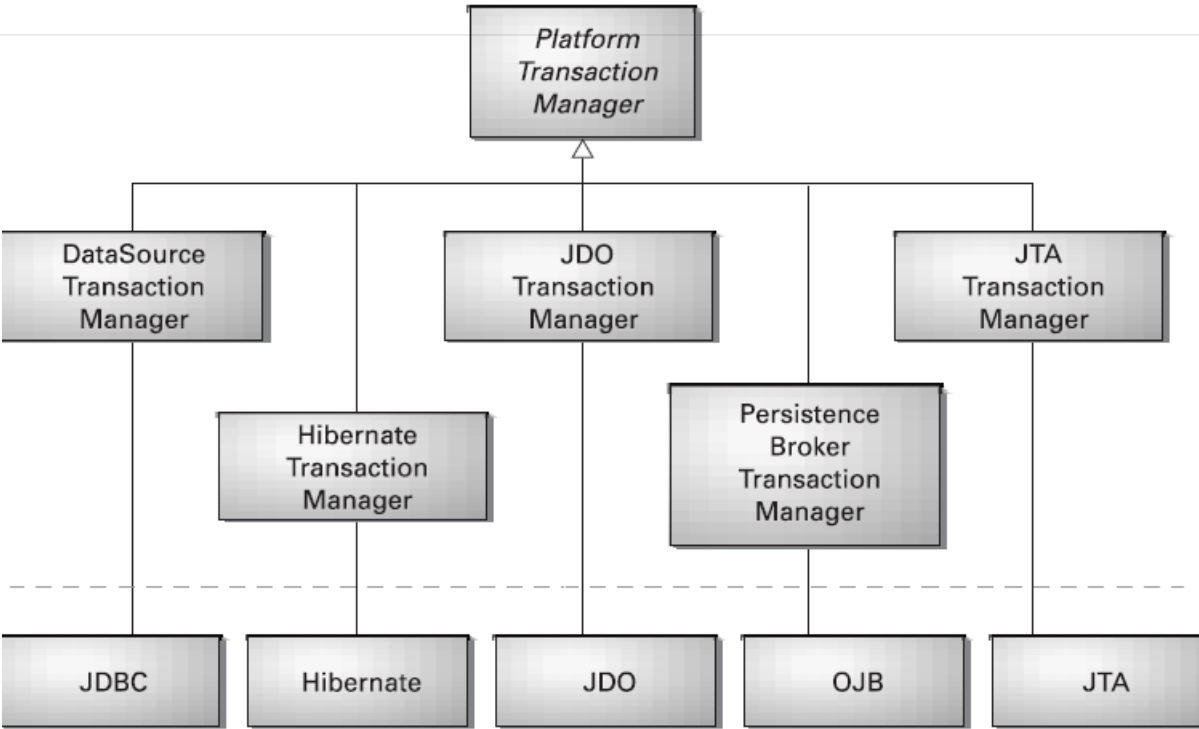
Spring对事务管理的支持

与EJB类似，Spring提供了对 编码式 和 声明式 事务管理的支持。但是，Spring对事务管理的能力远远超过EJB。这里就不详细介绍编码式事务和声明式事务的区别了。有兴趣的读者可以自行Google。

Spring对事务管理是通过事务管理器来实现的。Spring提供了许多内置事务管理器实现：

事务管理器 (org.springframework.*)	使用场景
DataSourceTransactionManager	数据源事务管理器，提供对单个 javax.sql.DataSource 事务管理，用于Spring JDBC抽象框架、iBATIS或MyBatis框架的事务管理；
JdoTransactionManager	提供对单个 javax.jdo.PersistenceManagerFactory 事务管理，用于集成JDO框架时的事务管理；
JpaTransactionManager	提供对单个 javax.persistence.EntityManagerFactory 事务支持，用于集成JPA实现框架时的事务管理；
HibernateTransactionManager	提供对单个 org.hibernate.SessionFactory 事务支持，用于集成Hibernate框架时的事务管理；该事务管理器只支持Hibernate3+版本，且Spring3.0+版本只支持Hibernate 3.2+版本；
JtaTransactionManager	提供对分布式事务管理的支持，并将事务管理委托给Java EE应用服务器事务管理器；
OC4JjtaTransactionManager	Spring提供的对OC4J10.1.3+应用服务器事务管理器的适配器，此适配器用于对应用服务器提供的高级事务的支持；
WebSphereUowTransactionManager	Spring提供的对WebSphere 6.0+应用服务器事务管理器的适配器，此适配器用于对应用服务器提供的高级事务的支持；
WebLogicJtaTransactionManager	Spring提供的对WebLogic 8.1+应用服务器事务管理器的适配器，此适配器用于对应用服务器提供的高级事务的支持。

Spring's Transaction Managers



Platform-Specific Transaction Implementations

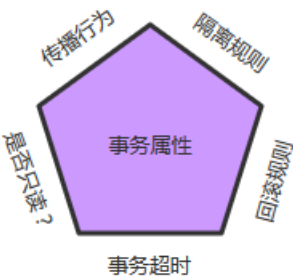
以上就是 Spring 中支持使用的事务管理器，一般我们比较常用的就是 `HibernateTransactionManager` 和 `DataSourceTransactionManager`。我们在使用事务的时候要声明要使用哪种事务管理器。如：

```
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource" />
</bean>
```

事务属性

spring中，声明事务是通过事务属性来定义的。事务属性描述了事务策略如何应用到方法上事务属性包含5个方面：

- 传播行为
- 隔离级别
- 回滚规则
- 事务超时
- 是否只读



这里简单介绍一下这五个属性。

传播行为

传播行为定义了客户端与被调用方法之间的事务边界，即传播规则回答了这样的一个问题，新的事务应该被启动还是挂起，或者方法是否要在事务环境中运行。（后面会有单独的文章介绍该属性）

隔离级别

隔离级别定义了一个事务可能受其他并发事务影响的程度。多事务并发可能会导致脏读、幻读、不可重复读等各种读现象。（具体参考：[数据库的读现象浅析 \(http://www.hollischuang.com/archives/900\)](http://www.hollischuang.com/archives/900)）

- ISOLATION_DEFAULT:使用后端数据库默认的规则
- ISOLATION_READ_UNCOMMITTED:允许读取尚未提交的数据变更，可能会导致脏读，幻读或不可重复读
- ISOLATION_READ_COMMITTED:允许读取并发事务已经提交的数据，可以防止脏读，但是幻读或不可重复读仍有可能发生
- ISOLATION_REPEATABLE_READ:对同意字段的多次读取结果是一致的，除非数据是被本事务自己所修改，看阻止脏读和不可重复读，但幻读仍有可能发生
- ISOLATION_SERIALIZABLE:完全服从ACID的隔离级别，确保阻止脏读，不可重复读以及幻读，这是最慢的数据隔离级别

（具体参考：[深入分析事务的隔离级别 \(http://www.hollischuang.com/archives/943\)](http://www.hollischuang.com/archives/943)）

是否只读

如果事务只对后端的数据库进行读操作，数据库可以利用事务ID只读特性来进行一些特定的优化。通过将事务设置为只读，你就可以给数据库一个机会，让他应用它认为合适的优化措施。因为是否只读是在事务启动的时候由数据库实施的，所以只有对那些具备可能启动一个新事务的传播行为（PROPAGATION_REQUIRED , PROPAGATION_REQUIRED_NEW , PROPAGATION_NESTED ）的方法来说，才有意义。

事务超时

为了使应用程序很好地运行，事务不能运行太长时间。因为超时时钟会在事务开始时启动，所以只有对那些具备可能启动一个新事务的传播行为（PROPAGATION_REQUIRED , PROPAGATION_REQUIRED_NEW , PROPAGATION_NESTED ）的方法来说，才有意义。

事务回滚

事务回滚规则定义了哪些异常会导致事务回滚而哪些不会。默认情况下，事务只有在遇到运行时期异常才回滚，而在遇到检查型异常时不会回滚。

配置方式

事务属性的配置方式通过以下关键字来指定：

关键字	含义
isolation	指定事务的隔离级别
propagation	定义事务的传播规则
read-only	指定事务为只读

关键字	含义
rollback-for	rollback-for指定事务对哪些检查型异常应当回滚而不提交
no-rollback-for	no-rollback-for指定事务对哪些异常应当继续执行而不回滚
timeout	对于长时间运行的事务定义超时时间

XML中事务属性的配置方式如下：

```
<tx:advice id="txAdvice" transactionmanager="transactionManager">
    <tx:attributes>
        <tx:method name="*" propagation="REQUIRED" read-only="true"/>
    </tx:attributes>
</tx:advice>
```

注解中事务属性的配置方式如下：

```
@Transaction(propagation=Propagation.REQUIRED,readOnly=true)
public void add(String username){
    //...
}
```

总结

事务是企业应用开发中很重要的组成部分，他让软件变得更加健壮。他保证了全有或全无的操作。

Spring同时支持编码式和声明式事务管理，无论使用哪种方式进行事务管理，都应该知道与事务相关的五个属性。

参考资料

《Spring 实战》(http://s.click.taobao.com/t?e=m%3D2%26s%3DE%2FT2x4dEeDAcQipKwQzePOeEDrYVVa64K7Vc7tEgwiFRAdhuf14FMVdReslR4jvK5x%2BIUIGKNpWOI%2BaSJ1gBKBqqRJv926uOeMnzhByHij7AJ%2FO89eZa8aU8rDUeBFLayGckMejm0gQcWu95Lkbq3EqY%2Bakgpmw&pvid=50_121.0.29.201_320188_1462592359061).

spring_事务属性(http://www.lai18.com/content/4533758.html)



摄于 2016.04.16 @江郎山

AD：【公告】关注Hollis公众号，回复『资源』，获取2000G架构师晋级资料 (<http://www.hollischuang.com/wp-content/uploads/2018/10/1-1.png>)

👍 赞(1)

如未加特殊说明，此网站文章均为原创，转载必须注明出处。HollisChuang's Blog (<https://www.hollischuang.com>) » Spring的事务管理机制 (<https://www.hollischuang.com/archives/1489>)

分享到： 更多 (0)

标签： 事务 (<https://www.hollischuang.com/archives/tag/%e4%ba%8b%e5%8a%a1>)

上一篇
设计模式（九）——建造者模式
(<https://www.hollischuang.com/archives/1477>)

下一篇
设计模式（十一）——适配器模式
(<https://www.hollischuang.com/archives/1524>)

相关推荐

- Java中的事务——JDBC事务和JTA事务 (<https://www.hollischuang.com/archives/1658>)
- 深入分析事务的隔离级别 (<https://www.hollischuang.com/archives/943>)

- 深入理解乐观锁与悲观锁 (<https://www.hollischuang.com/archives/934>)
- MySQL中的共享锁与排他锁 (<https://www.hollischuang.com/archives/923>)
- MySQL中的行级锁,表级锁,页级锁 (<https://www.hollischuang.com/archives/914>)
- 数据库的锁机制 (<https://www.hollischuang.com/archives/909>)
- 数据库的读现象浅析 (<https://www.hollischuang.com/archives/900>)
- 彻底理解数据库事务 (<https://www.hollischuang.com/archives/898>)

评论 1



期待你的评论～

提交评论

昵称	昵称 (必填)
邮箱	邮箱 (必填)
网址	网址



确实不错，这个要实话实说！#1

一元营销 (<http://www.268595.zmmaa.com/>) 3年前 (2016-05-10) 回复

HollisChuang's Blog

联系我 (http://mail.qq.com/cgi-bin/qm_share?t=qm_mailme&email=-JSTkJCVj5_UiZ2Sm7yNjdKfk5E)

关于我 (/sample-page)