Spring 多数据源 @Transactional 注解事务管理

原创 2017年12月21日 15:38:30

```
在 Spring, MyBatis 下两个数据源,通过 @Transactional 注解 配置简单的事务管理 spring-mybatis.xml
```

```
<!-- 数据库连接池 -->
<bean id="dataSource" class="com.alibaba.druid.pool.DruidDataSource" init-method="init" destroy-method="close">
  property name="driverClassName" value="${jdbc.driver}" />
  property name="url" value="${jdbc.url}" />
  property name="username" value="${jdbc.username}" />
  property name="password" value="${jdbc.password}"
  property name="initialSize" value="10" />
  property name="minIdle" value="50" />
  property name="maxActive" value="100" />
  property name="maxWait" value="60000" />
  property name="useUnfairLock" value="true" />
  property name="timeBetweenEvictionRunsMillis" value="300000" />
  property name="minEvictableIdleTimeMillis" value="600000" />
  property name="validationQuery" value="select 1" />
  property name="testWhileIdle" value="true" />
  property name="testOnBorrow" value="true" />
  property name="poolPreparedStatements" value="true" />
  property name="maxOpenPreparedStatements" value="200" />
  \label{lem:connectionSize} $$ \operatorname{property name="maxPoolPreparedStatementPerConnectionSize" value="20"/} $$
<!-- 让spring管理sqlsessionfactory 使用mybatis和spring整合包中的 -->
<!-- 数据库连接池 -->
  property name="dataSource" ref="dataSource" />
  <!-- 加载mybatis的全局配置文件 -->
  property name="configLocation" value="classpath:mybatis/SqlMapConfig.xml" />
  <!-- 自动扫描mapping.xml文件 -->
    property name="mapperLocations" value="classpath:mybatis/mapping/*.xml" />
</bean>
<!-- DAO接口所在包名, Spring会自动查找其下的类 -->
cproperty name="basePackage" value="com.***.***.dao" /
  property name="sqlSessionFactoryBeanName" value="sqlSessionFactory" />
</bean>
<!-- 设定transactionManager -->
<bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  property name="dataSource" ref="dataSource" />
</bean>
<!-- 支持 @Transactional 标记 -->
<tx:annotation-driven transaction-manager="transactionManager" proxy-target-class="true" />
<!-- 数据库连接池 -->
<!--略写-->
</bean>
<!-- 让spring管理sqlsessionfactory 使用mybatis和spring整合包中的 -->
property name="dataSource" ref="dataSourceTwo" />
```

```
<!--略写-->
</bean>
<!-- DAO接口所在包名, Spring会自动查找其下的类 -->
property name="sqlSessionFactoryBeanName" value="sqlSessionFactoryTwo" />
</bean>
<!-- 设定transactionManager -->
property name="dataSource" ref="dataSourceTwo" />
</hean>
<!-- 支持 @Transactional 标记 -->
<tx:annotation-driven transaction-manager="transactionManagerTwo" proxy-target-class="true" />
上面配置了两个数据源(dataSource, dataSourceTwo),两个事务管理器(transactionManager,tra
nsactionManagerTwo)
@Transactional 事务注解
1)第一个数据源,使用第一个事务管理器
@Override
@Transactional(value = "transactionManager", readOnly = false)
public void insert {
2) 第二个数据源,使用第二个事务管理器
@Override
@Transactional(value = "transactionManagerTwo", readOnly = false)
public void insert {
3) 不使用 value 或者 @Transactional 标记, 会使用 哪个数据源 及起哪个事务?
@Override
@Transactional
public void insert {
@Override
public void insert {
像这种不标明的情况,会使用默认事务管理器(transactionManager),所以只有一个事务管理器的时
候,我们一般不写 value
```

引出:一条SQL语句是否需要事务

事务的存在意义:当我们在执行数据库操作时,例如:insert,update,delete ,当发生错误导致执行失败时,我们需要对整个操作进行回滚。

当我们只有一个操作如 insert,我们是不需要搞个事务跟踪的, 因为一条语句就是一个事务,它自己本身就已经有事务 $7 \cdot 1 \cdot 1$

就算这条 insert 语句执行失败,它自然不会插入成功,同时也不会对其它操作造成影响。

按我理解,事务应该用在执行多条语句,并且语句之间会相互影响的时候才需要,事务是用在多个语句保证同时成功的时候用,单个语句不建议用

需要注意的是,多个数据库的这种配置是不支持分布式事务的,也就是同一个事务中,不能操作多个数据库。

如果要实现分布式事务可参考:

Spring分布式事务实现

spring多数据源配置

Spring, MyBatis 多数据源的配置和管理

版权声明:本文为博主原创文章,未经博主允许不得转载。



发表你的评论