## Spring 下 mybatis 多数据源配置

介绍

本文描述了以 mybatis 作为持久层框架使用 Spring 的声明式事务时,如何配置多个数据源(即连接多个数据库),使用 Spring 的注解方式进行依赖的注入和事务的管理。并且利用 mybatis 的 spring 插件自动扫描和装配 Mapper 接口。

## 先来看一个 mybatis 的单数据源配置

```
<context:property-placeholder location="classpath:jdbc.properties" />
   <bean id="dataSource" class="org.logicalcobwebs.proxool.ProxoolDataSource">
       cproperty name="alias" value="${alias}" />
       cproperty name="driver" value="${driver}" />
       cproperty name="driverUrl" value="${driverUrl}" />
       cproperty name="user" value="${user}" />
       cproperty name="password" value="${password}" />
       cproperty name="prototypeCount" value="${prototype-count}" />
       property name="simultaneousBuildThrottle" value="${simultaneous-build-throttle}" />
       cproperty name="trace" value="${trace}" />
   </bean>
   <bean id="transactionManager"</pre>
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
       cproperty name="dataSource" ref="dataSource" />
   </bean>
   <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
       cproperty name="dataSource" ref="dataSource" />
       roperty name="configLocation" value="classpath:mybatis-config.xml">
   </bean>
   <tx:annotation-driven transaction-manager="transactionManager" />
   <!-- 自动扫描mapper接口 -->
   <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
```

## 说明

- 1. 数据库连接池使用 Proxool, 首先配置的就是 Proxool 的数据源
- 2. 接下来配置 Spring 的声明式事务管理,这里我们使用全注解+自动装配的方式进行 Bena 和事务声明。
- 3. mybatis 配置为自动扫描 Maper 接口,只要指定 Mapper 接口所在的包,需要注意的是, Mapper 接口对应的映射文件(.xml)也要在这个包里,且名字和接口的名字一致。使用多个数据源时, mybatis 的文档表示不能使用自动扫描和自动装配,但实际使用时发现还是可以的。看一个配置文件先:

```
<context:property-placeholder location="classpath:jdbc.properties" />
<!-- proxool连接池 -->
<bean id="dataSource" class="org.logicalcobwebs.proxool.ProxoolDataSource">
   cproperty name="alias" value="${alias}" />
   cproperty name="driver" value="${driver}" />
   cproperty name="driverUrl" value="${driverUrl}" />
   cproperty name="user" value="${user}" />
   cproperty name="password" value="${password}" />
   cproperty name="prototypeCount" value="${prototype-count}" />
   cproperty name="trace" value="${trace}" />
</bean>
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
   cproperty name="dataSource" ref="dataSource" />
   cproperty name="configLocation" value="classpath:mybatis-config.xml"/>
</bean>
<bean id="transactionManager"</pre>
   class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
   cproperty name="dataSource" ref="dataSource" />
</bean>
```

```
<!-- 注解式事务管理 -->
   <tx:annotation-driven transaction-manager="transactionManager" />
   <!-- 自动扫描 mybatis mapper接口 -->
   <bean name="mapperScannerConfigurer"</pre>
class="org.mybatis.spring.mapper.MapperScannerConfigurer">
       cproperty name="basePackage" value="rhino.persistence.mapper" />
       cproperty name="sqlSessionFactory" ref="sqlSessionFactory" />
   </bean>
   <!-- 配置新的的数据源 -->
   <bean id="dataSource_1" class="org.logicalcobwebs.proxool.ProxoolDataSource">
       cproperty name="alias" value="${db1.alias}" />
       cproperty name="driver" value="${driver}" />
       cproperty name="driverUrl" value="${db1.driverUrl}" />
       cproperty name="user" value="${db1.user}" />
       cproperty name="password" value="${db1.password}" />
       cproperty name="maximumConnectionCount" value="${maximum-connection-count}" />
       cproperty name="prototypeCount" value="${prototype-count}" />
       cproperty name="trace" value="${trace}" />
   </bean>
   <bean id="sqlSessionFactory_1" class="org.mybatis.spring.SqlSessionFactoryBean">
       cproperty name="dataSource" ref="dataSource_1" />
       cproperty name="configLocation" value="classpath:mybatis-config-db1.xml"/>
   </bean>
   <bean id="transactionManager 1"</pre>
       class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
       cproperty name="dataSource" ref="dataSource_1" />
   </bean>
   <tx:annotation-driven transaction-manager="transactionManager 1" />
   <!-- 自动扫描 mybatis mapper接口 -->
   <bean name="mapperScannerConfigurer_1"</pre>
class="org.mybatis.spring.mapper.MapperScannerConfigurer">
       cproperty name="basePackage" value=rhino.persistence.db1" />
       cproperty name="sqlSessionFactory" ref="sqlSessionFactory_1" />
   </bean>
```

```
<!-- spring beans 组件扫描 --> <context:component-scan base-package="rhino.domain" />
```

## 说明

- 1. 需要配置新的数据源、事务管理器、sqlSession 工厂
- 2. mybatis 配置文件也要独立出来,即 2 个数据源应该有 2 个 mybatis 配置文件,互相 之间是独立的 注意配置 sqlSessionFactory\_1 时注入的 mybatis-config-db1.xml
- 3. Mapper 接口要放在一个新的包里
- 4. 2个以上的数据源同样配置