数据库连接池c3p0和dbcp

现在常用的开源数据连接池主要有c3p0、dbcp和proxool三种，其中：

* hibernate开发组推荐使用c3p0;
* spring开发组推荐使用dbcp(dbcp连接池有weblogic连接池同样的问题，就是强行关闭连接或数据库重启后，无法reconnect，告诉连接被重置，这个设置可以解决);
* hibernate in action推荐使用c3p0和proxool;

**dbcp所需jar**：commons-dbcp.jar、commons-pool.jar

**c3p0所需jar**：c3p0-0.9.2.1.jar mchange-commons-java-0.2.3.4.jar

**C3P0简介**：C3P0是一个开源的JDBC连接池，它实现了数据源和JNDI绑定，支持JDBC3规范和JDBC2的标准扩展。目前使用它的开源项目有Hibernate，Spring等。

**dbcp简介**： DBCP(DataBase connection pool),数据库连接池。是 apache 上的一个 java 连接池项目，也是 tomcat 使用的连接池组件。单独使用dbcp需要3个包：common-dbcp.jar,common-pool.jar,common-collections.jar由于建立数据库连接是一个非常耗时耗资源的行为，所以通过连接池预先同数据库建立一些连接，放在内存中，应用程序需要建立数据库连接时直接到连接池中申请一个就行，用完后再放回去。

**c3p0与dbcp区别**：dbcp没有自动的去回收空闲连接的功能 c3p0有自动回收空闲连接功能 。两者主要是对数据连接的处理方式不同！C3P0提供最大空闲时间，DBCP提供最大连接数。 前者当连接超过最大空闲连接时间时，当前连接就会被断掉。DBCP当连接数超过最大连接数时，所有连接都会被断开。

# 配置方法

|  |
| --- |
| **<!-- 配置dbcp数据源 -->**  <bean id="dataSource2" destroy-method="close" class="org.apache.commons.dbcp.BasicDataSource">  <property name="driverClassName" value="${jdbc.driverClassName}"/>  <property name="url" value="${jdbc.url}"/>  <property name="username" value="${jdbc.username}"/>  <property name="password" value="${jdbc.password}"/>  <!-- 池启动时创建的连接数量 -->  <property name="initialSize" value="5"/>  <!-- 同一时间可以从池分配的最多连接数量。设置为0时表示无限制。 -->  <property name="maxActive" value="30"/>  <!-- 池里不会被释放的最多空闲连接数量。设置为0时表示无限制。 -->  <property name="maxIdle" value="20"/>  <!-- 在不新建连接的条件下，池中保持空闲的最少连接数。 -->  <property name="minIdle" value="3"/>  <!-- 设置自动回收超时连接 -->  <property name="removeAbandoned" value="true" />  <!-- 自动回收超时时间(以秒数为单位) -->  <property name="removeAbandonedTimeout" value="200"/>  <!-- 设置在自动回收超时连接的时候打印连接的超时错误 -->  <property name="logAbandoned" value="true"/>  <!-- 等待超时以毫秒为单位，在抛出异常之前，池等待连接被回收的最长时间（当没有可用连接时）。设置为-1表示无限等待。 -->  <property name="maxWait" value="100"/>  </bean>    **<!-- 配置c3p0数据源 -->**  <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">  <property name="jdbcUrl" value="${jdbc.url}" />  <property name="driverClass" value="${jdbc.driverClassName}" />  <property name="user" value="${jdbc.username}" />  <property name="password" value="${jdbc.password}" />  <!--连接池中保留的最大连接数。Default: 15 -->  <property name="maxPoolSize" value="100" />  <!--连接池中保留的最小连接数。-->  <property name="minPoolSize" value="1" />  <!--初始化时获取的连接数，取值应在minPoolSize与maxPoolSize之间。Default: 3 -->  <property name="initialPoolSize" value="10" />  <!--最大空闲时间,60秒内未使用则连接被丢弃。若为0则永不丢弃。Default: 0 -->  <property name="maxIdleTime" value="30" />  <!--当连接池中的连接耗尽的时候c3p0一次同时获取的连接数。Default: 3 -->  <property name="acquireIncrement" value="5" />  <!--JDBC的标准参数，用以控制数据源内加载的PreparedStatements数量。但由于预缓存的statements  属于单个connection而不是整个连接池。所以设置这个参数需要考虑到多方面的因素。  如果maxStatements与maxStatementsPerConnection均为0，则缓存被关闭。Default: 0-->  <property name="maxStatements" value="0" />    <!--每60秒检查所有连接池中的空闲连接。Default: 0 -->  <property name="idleConnectionTestPeriod" value="60" />    <!--定义在从数据库获取新连接失败后重复尝试的次数。Default: 30 -->  <property name="acquireRetryAttempts" value="30" />    <!--获取连接失败将会引起所有等待连接池来获取连接的线程抛出异常。但是数据源仍有效  保留，并在下次调用getConnection()的时候继续尝试获取连接。如果设为true，那么在尝试  获取连接失败后该数据源将申明已断开并永久关闭。Default: false-->  <property name="breakAfterAcquireFailure" value="true" />    <!--因性能消耗大请只在需要的时候使用它。如果设为true那么在每个connection提交的  时候都将校验其有效性。建议使用idleConnectionTestPeriod或automaticTestTable  等方法来提升连接测试的性能。Default: false -->  <property name="testConnectionOnCheckout" value="false" />  </bean> |