[Spring中配置数据源的4种形式](http://blog.csdn.net/orclight/article/details/8616103)

2013-02-27 02:16 11121人阅读 [评论](http://blog.csdn.net/orclight/article/details/8616103#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/orclight/article/details/8616103#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

数据源（1） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录

1. [spring自带的数据源](http://blog.csdn.net/orclight/article/details/8616103#t0)
2. [DBCP数据源](http://blog.csdn.net/orclight/article/details/8616103#t1)
3. [C3P0数据源](http://blog.csdn.net/orclight/article/details/8616103#t2)
4. [JNDI数据源](http://blog.csdn.net/orclight/article/details/8616103#t3)

不管采用何种持久化技术，都需要定义数据源。Spring中提供了4种不同形式的数据源配置方式：**spring自带的数据源(DriverManagerDataSource)，DBCP数据源，C3P0数据源,JNDI数据源。**

**1.spring自带的数据源**

**DriverManagerDataSource**

XML代码：

**[html]** [view plain](http://blog.csdn.net/orclight/article/details/8616103) [copy](http://blog.csdn.net/orclight/article/details/8616103)

1. **<bean** id="dataSource"
2. class="org.springframework.jdbc.datasource.DriverManagerDataSource"**>**
3. **<property** name="driverClassName" value="oracle.jdbc.driver.OracleDriver" **/>**
4. **<property** name="url" value="jdbc:oracle:thin:@172.19.34.6:1521:ORCL" **/>**
5. **<property** name="username" value="orclight" **/>**
6. **<property** name="password" value="123456" **/>**
7. **</bean>**

**2.DBCP数据源**

    DBCP的配置依赖于2个jar包**commons-dbcp.jar，commons-pool.jar。**

XML代码：

**[html]** [view plain](http://blog.csdn.net/orclight/article/details/8616103) [copy](http://blog.csdn.net/orclight/article/details/8616103)

1. **<bean** id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
2. destroy-method="close"**>**
3. **<property** name="driverClassName" value="oracle.jdbc.driver.OracleDriver" **/>**
4. **<property** name="url" value="jdbc:oracle:thin:@172.19.34.6:1521:ORCL" **/>**
5. **<property** name="username" value="orclight" **/>**
6. **<property** name="password" value="123456" **/>**
7. **</bean>**

上面代码的解释：

BasicDataSource提供了close()方法关闭数据源，所以必须设定destroy-method=”close”属性， 以便Spring容器关闭时，数据源能够正常关闭。除以上必须的数据源属性外，还有一些常用的属性：   
    defaultAutoCommit：设置从数据源中返回的连接是否采用自动提交机制，默认值为 true；   
    defaultReadOnly：设置数据源是否仅能执行只读操作， 默认值为 false；   
    maxActive：最大连接数据库连接数，设置为0时，表示没有限制；   
    maxIdle：最大等待连接中的数量，设置为0时，表示没有限制；   
    maxWait：最大等待秒数，单位为毫秒， 超过时间会报出错误信息；   
    validationQuery：用于验证连接是否成功的查询SQL语句，SQL语句必须至少要返回一行数据， 如你可以简单地设置为：“select count(\*) from user”；   
    removeAbandoned：是否自我中断，默认是 false ；   
    removeAbandonedTimeout：几秒后数据连接会自动断开，在removeAbandoned为true，提供该值；   
    logAbandoned：是否记录中断事件， 默认为 false；

**3.C3P0数据源**

     C3P0是一个开放源代码的JDBC数据源实现项目，C3P0依赖于jar包**c3p0.jar**。

XML代码：

**[html]** [view plain](http://blog.csdn.net/orclight/article/details/8616103) [copy](http://blog.csdn.net/orclight/article/details/8616103)

1. **<bean** id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"
2. destroy-method="close"**>**
3. **<property** name="driverClass" value=" oracle.jdbc.driver.OracleDriver "**/>**
4. **<property** name="jdbcUrl" value="jdbc:oracle:thin:@172.19.34.6:1521:ORCL"**/>**
5. **<property** name="user" value="orclight"**/>**
6. **<property** name="password" value="123456"**/>**
7. **</bean>**

ComboPooledDataSource和BasicDataSource一样提供了一个用于关闭数据源的close()方法，这样我们就可以保证Spring容器关闭时数据源能够成功释放。

    C3P0拥有比DBCP更丰富的配置属性，通过这些属性，可以对数据源进行各种有效的控制：  
    acquireIncrement：当连接池中的连接用完时，C3P0一次性创建新连接的数目；  
    acquireRetryAttempts：定义在从数据库获取新连接失败后重复尝试获取的次数，默认为30；  
    acquireRetryDelay：两次连接中间隔时间，单位毫秒，默认为1000；  
    autoCommitOnClose：连接关闭时默认将所有未提交的操作回滚。默认为false；  
    automaticTestTable： C3P0将建一张名为Test的空表，并使用其自带的查询语句进行测试。如果定义了这个参数，那么属性preferredTestQuery将被忽略。你 不能在这张Test表上进行任何操作，它将中为C3P0测试所用，默认为null；  
    breakAfterAcquireFailure：获取连接失败将会引起所有等待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。默认为 false；  
    checkoutTimeout：当连接池用完时客户端调用getConnection()后等待获取新连接的时间，超时后将抛出SQLException，如设为0则无限期等待。单位毫秒，默认为0；  
    connectionTesterClassName： 通过实现ConnectionTester或QueryConnectionTester的类来测试连接，类名需设置为全限定名。默认为 com.mchange.v2.C3P0.impl.DefaultConnectionTester；   
    idleConnectionTestPeriod：隔多少秒检查所有连接池中的空闲连接，默认为0表示不检查；  
    initialPoolSize：初始化时创建的连接数，应在minPoolSize与maxPoolSize之间取值。默认为3；  
    maxIdleTime：最大空闲时间，超过空闲时间的连接将被丢弃。为0或负数则永不丢弃。默认为0；  
    maxPoolSize：连接池中保留的最大连接数。默认为15；  
    maxStatements：JDBC的标准参数，用以控制数据源内加载的PreparedStatement数量。但由于预缓存的Statement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。默认为0；  
    maxStatementsPerConnection：连接池内单个连接所拥有的最大缓存Statement数。默认为0；  
    numHelperThreads：C3P0是异步操作的，缓慢的JDBC操作通过帮助进程完成。扩展这些操作可以有效的提升性能，通过多线程实现多个操作同时被执行。默认为3；  
    preferredTestQuery：定义所有连接测试都执行的测试语句。在使用连接测试的情况下这个参数能显著提高测试速度。测试的表必须在初始数据源的时候就存在。默认为null；  
    propertyCycle： 用户修改系统配置参数执行前最多等待的秒数。默认为300；  
    testConnectionOnCheckout：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。默认为false；  
    testConnectionOnCheckin：如果设为true那么在取得连接的同时将校验连接的有效性。默认为false。

**4.JNDI数据源**

    如果应用配置在高性能的应用服务器（如WebLogic或Websphere,tomcat等）上，我们可能更希望使用应用服务器本身提供的数据源。应用服务器的数据源 使用JNDI开放调用者使用，Spring为此专门提供引用JNDI资源的**JndiObjectFactoryBean**类。

xml 代码：

**[html]** [view plain](http://blog.csdn.net/orclight/article/details/8616103) [copy](http://blog.csdn.net/orclight/article/details/8616103)

1. **<bean** id="dataSource" class="org.springframework.jndi.JndiObjectFactoryBean"**>**
2. **<property** name="jndiName" value="java:comp/env/jdbc/orclight"**/>**
3. **</bean>**

**[html]** [view plain](http://blog.csdn.net/orclight/article/details/8616103) [copy](http://blog.csdn.net/orclight/article/details/8616103)

1. **<beans** xmlns=http://www.springframework.org/schema/beans
2. xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
3. xmlns:jee=http://www.springframework.org/schema/jee
4. xsi:schemaLocation="http://www.springframework.org/schema/beans
5. http://www.springframework.org/schema/beans/spring-beans-2.0.xsd
6. http://www.springframework.org/schema/jee
7. http://www.springframework.org/schema/jee/spring-jee-2.0.xsd"**>**
8. **<jee:jndi-lookup** id="dataSource" jndi-name=" java:comp/env/jdbc/orclight"**/>**
9. **</beans>**

参考文章：

1.[Spring配置数据源四种方式](http://www.2cto.com/kf/201301/184061.html)   <http://www.2cto.com/kf/201301/184061.html>

2.[配置Spring数据源](http://www.cnblogs.com/200911/archive/2012/08/10/2631760.html)  <http://www.cnblogs.com/200911/archive/2012/08/10/2631760.html>

3.[Spring 多数据源配置方法](http://blog.csdn.net/nickbest85/article/details/5510466)  <http://blog.csdn.net/nickbest85/article/details/5510466>

4.[如何在spring框架中解决多数据源的问题](http://www.iteye.com/topic/72486)   <http://www.iteye.com/topic/72486>