

# 少平的博客人生

be honest, stay hungry.

博客园

首页

新随笔

联系

订阅

管理

随笔 - 2 文章 - 0 评论 - 7

## 常用数据库连接池 (DBCP、c3p0、Druid) 配置说明

### 1. 引言

#### 1.1 定义

数据库连接是一种关键的有限的昂贵的资源，这一点在多用户的网页应用程序中体现得尤为突出。对数据库连接的管理能显著影响到整个应用程序的伸缩性和健壮性，影响到程序的性能指标。数据库连接池正是针对这个问题提出来的。

数据库连接池负责分配、管理和释放数据库连接，它允许应用程序重复使用一个现有的数据库连接，而不是再重新建立一个；释放空闲时间超过最大空闲时间的数据库连接来避免因为没有释放数据库连接而引起的数据库连接遗漏。这项技术能明显提高对数据库操作的性能。

#### 1.2 参考资料

DBCP

下载地址：[http://commons.apache.org/proper/commons-dbcp/download\\_dbcp.cgi](http://commons.apache.org/proper/commons-dbcp/download_dbcp.cgi)

相关属性说明：<http://commons.apache.org/proper/commons-dbcp/configuration.html>

c3p0

下载地址：<http://sourceforge.net/projects/c3p0/>

相关属性说明：<http://www.mchange.com/projects/c3p0/>

Druid

下载地址：<http://repo1.maven.org/maven2/com/alibaba/druid/>

相关属性说明：<https://github.com/alibaba/druid/wiki>

### 2. 数据库连接池

#### 2.1 原理

连接池基本的思想是在系统初始化的时候，将数据库连接作为对象存储在内存中，当用户需要访问数据库时，并非建立一个新的连接，而是从连接池中取出一个已建立的空闲连接对象。使用完毕后，用户也并非将连接关闭，而是将连接放回连接池中，以供下一个请求访问使用。而连接的建立、断开都由连接池自身来管理。同时，还可以通过设置连接池的参数来控制连接池中的初始连接数、连接的上下限数以及每个连接的最大使用次数、最大空闲时间等等，也可以通过其自身的管理机制来监视数据库连接的数量、使用情况等。

#### 2.2 常见数据库连接池及其特点

在Java中开源的常用的数据库连接池有以下几种：

1) DBCP

DBCP是一个依赖Jakarta commons-pool对象池机制的数据库连接池。DBCP可以直接的在应用程序中使用，Tomcat的数据源使用的就是DBCP。

2) c3p0

c3p0是一个开放源代码的JDBC连接池，它在lib目录中与Hibernate一起发布，包括了实现jdbc3和jdbc2扩展规范说明的Connection 和Statement 池的DataSources 对象。

3) Druid

#### 公告

昵称：shaopiing

园龄：4年4个月

粉丝：11

关注：0

+加关注

2020年7月						
日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

#### 常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

#### 我的标签

Java SE(1)

Java Web(1)

数据库连接池(1)

#### 最新评论

1. Re:常用数据库连接池 (DBCP、c3p0、Druid) 配置说明  
@ 掌厝走天涯当然可以，我这个也是参考各种官方文档总结的。...

--shaopiing

2. Re:常用数据库连接池 (DBCP、c3p0、Druid) 配置说明  
six

--Arebirth

3. Re:常用数据库连接池 (DBCP、c3p0、Druid) 配置说明  
您好大哥,这篇文章我能复制到我的csnd中吗

--掌厝走天涯

4. Re:常用数据库连接池 (DBCP、c3p0、Druid) 配置说明  
哇塞,这也太全面了吧

--掌厝走天涯

5. Re:常用数据库连接池 (DBCP、c3p0、Druid) 配置说明  
你这个确实差不多是抄完的

--越秃越强1

阿里出品，淘宝和支付宝专用数据库连接池，但它不仅仅是一个数据库连接池，它还包含一个ProxyDriver，一系列内置的JDBC组件库，一个SQL Parser。支持所有JDBC兼容的数据库，包括Oracle、MySql、Derby、Postgresql、SQL Server、H2等等。

Druid针对Oracle和MySql做了特别优化，比如Oracle的PS Cache内存占用优化，MySql的ping检测优化。

Druid提供了MySql、Oracle、Postgresql、SQL-92的SQL的完整支持，这是一个手写的高性能SQL Parser，支持Visitor模式，使得分析SQL的抽象语法树很方便。

简单SQL语句用时10微秒以内，复杂SQL用时30微秒。

通过Druid提供的SQL Parser可以在JDBC层拦截SQL做相应处理，比如说分库分表、审计等。Druid防御SQL注入攻击的WallFilter就是通过Druid的SQL Parser分析语义实现的。

### 3. 主要配置说明

连接池配置大体可以分为基本配置、关键配置、性能配置等主要配置。

#### 3.1 基本配置

基本配置是指连接池进行数据库连接的四个基本必需配置：

传递给JDBC驱动的用于连接数据库的用户名、密码、URL以及驱动类名。

	DBCP	c3p0	Druid
用户名	username	user	username
密码	password	password	password
URL	url	jdbcUrl	jdbcUrl
驱动类名	driverClassName	driverClass	driverClassName

注:在Druid连接池的配置中，driverClassName可配可不配，如果不配置会根据url自动识别dbType(数据库类型)，然后选择相应的driverClassName。

#### 3.2 关键配置

为了发挥数据库连接池的作用，在初始化时将创建一定数量的数据库连接放到连接池中，这些数据库连接的数量是由最小数据库连接数来设定的。无论这些数据库连接是否被使用，连接池都将一直保证至少拥有这么多的连接数量。连接池的最大数据库连接数量限定了这个连接池能占有的最大连接数，当应用程序向连接池请求的连接数超过最大连接数量时，这些请求将被加入到等待队列中。

最小连接数：

是数据库一直保持的数据库连接数，所以如果应用程序对数据库连接的使用量不大，将有大量的数据库资源被浪费。

初始化连接数：

连接池启动时创建的初始化数据库连接数量。

最大连接数：

是连接池能申请的最大连接数，如果数据库连接请求超过此数，后面的数据库连接请求被加入到等待队列中。

最大等待时间：

当没有可用连接时，连接池等待连接被归还的最大时间，超过时间则抛出异常，可设置参数为0或者负数使得无限等待(根据不同连接池配置)。

	DBCP	c3p0	Druid
最小连接数	minIdle(0)	minPoolSize(3)	minIdle(0)
初始化连接数	initialSize(0)	initialPoolSize(3)	initialSize(0)

#### 阅读排行榜

- 1. 常用数据库连接池 (DBCP、c3p0、Druid) 配置说明(45269)
- 2. FindBugs：Java 静态代码检查(3486)

#### 推荐排行榜

- 1. 常用数据库连接池 (DBCP、c3p0、Druid) 配置说明(14)

最大连接数	maxTotal(8)	maxPoolSize(15)	maxActive(8)
最大等待时间	maxWaitMillis(毫秒)	maxIdleTime(0秒)	maxWait(毫秒)

注1:在DBCP连接池的配置中，还有一个maxIdle的属性，表示最大空闲连接数，超过的空闲连接将被释放，默认值为8。对应的该属性在Druid连接池已不再使用，配置了也没有效果，c3p0连接池则没有对应的属性。

注2：数据库连接池在初始化的时候会创建initialSize个连接，当有数据库操作时，会从池中取出一个连接。如果当前池中正在使用的连接数等于maxActive，则会等待一段时间，等待其他操作释放掉某一个连接，如果这个等待时间超过了maxWait，则会报错；如果当前正在使用的连接数没有达到maxActive，则判断当前是否空闲连接，如果有则直接使用空闲连接，如果没有则新建建立一个连接。在连接使用完毕后，不是将其物理连接关闭，而是将其放入池中等待其他操作复用。

### 3.3 性能配置

预缓存设置:

即是PSCache，PSCache对支持游标的数据库性能提升巨大，比如说oracle。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。

单个连接拥有的最大缓存数：要启用PSCache，必须配置大于0，当大于0时，poolPreparedStatements自动触发修改为true。在Druid中，不会存在Oracle下PSCache占用内存过多的问题，可以把这个数值配置大一些，比如说100

	DBCP	c3p0	Druid
开启缓存功能	poolPreparedStatements	maxStatements	poolPreparedStatements
单个连接拥有的最大缓存数	maxOpenPrepared-Statements	maxStatementsPer-Connection	maxOpenPrepared-Statements

连接有效性检测设置:

连接池内部有机制判断，如果当前的总的连接数少于minIdle，则会建立新的空闲连接，以保证连接数得到minIdle。如果当前连接池中某个连接在空闲了timeBetweenEvictionRunsMillis时间后任然没有使用，则被物理性的关闭掉。有些数据库连接的时候有超时限制（mysql连接在8小时后断开），或者由于网络中断等原因，连接池的连接会出现失效的情况，这时候设置一个testWhileIdle参数为true，可以保证连接池内部定时检测连接的可用性，不可用的连接会被抛弃或者重建，最大情况的保证从连接池中得到的Connection对象是可用的。当然，为了保证绝对的可用性，你也可以使用testOnBorrow为true（即在获取Connection对象时检测其可用性），不过这样会影响性能。

	DBCP	c3p0	Druid
申请连接检测	testOnBorrow	testConnectionOnCheckin	testOnBorrow
是否超时检测	testWhileIdle		testWhileIdle
空闲时间	timeBetweenEvictionRunsMillis	idleConnectionTestPeriod	timeBetweenEvictionRunsMillis
校验用sql语句	validationQuery	preferredTestQuery	validationQuery
归还连接检测	testOnReturn	testConnectionOnCheckout	testOnReturn

超时连接关闭设置:

removeAbandoned参数，用来检测到当前使用的连接是否发生了连接泄露，所以在代码内部就假定如果一个连接建立连接的时间很长，则将其认定为泄露，继而强制将其关闭掉。

	DBCP	c3p0	Druid
是否超时关闭连接	removeAbandoned	breakAfterAcquireFailure	removeAbandoned
超时时间	removeAbandonedTimeout	checkoutTimeout	removeAbandonedTimeout
是否记录日志	logAbandoned		logAbandoned

c3p0重连设置:

设置获取连接失败后，是否重新连接以及间隔时间。

	DBCP	c3p0	Druid
重连次数		acquireRetryAttempts	
间隔时间		acquireRetryDelay	

## 4. 配置详解

### 4.1 DBCP 属性说明表

属性(Parameter)	默认值(Default)	描述(Description)
username		传递给JDBC驱动的用于建立连接的用户名 (The connection username to be passed to our JDBC driver to establish a connection.)
password		传递给JDBC驱动的用于建立连接的密码(The connection password to be passed to our JDBC driver to establish a connection.)
url		传递给JDBC驱动的用于建立连接的URL(The connection URL to be passed to our JDBC driver to establish a connection.)
driverClassName		使用的JDBC驱动的完整有效的java 类名 (The fully qualified Java class name of the JDBC driver to be used.)
defaultAutoCommit	driver default	连接池创建的连接的默认的auto-commit状态,没有设置则不会自动提交(The default auto-commit state of connections created by this pool. If not set then the setAutoCommit method will not be called.)
initialSize	0	初始化连接:连接池启动时创建的初始化连接数量(The initial number of connections that

		are created when the pool is started.
<b>maxTotal</b>	<b>8</b>	最大活动连接:连接池在同一时间能够分配的最大活动连接的数量, 如果设置为非正数则表示不限制(The maximum number of active connections that can be allocated from this pool at the same time, or negative for no limit.)
<b>maxIdle</b>	<b>8</b>	最大空闲连接:连接池中容许保持空闲状态的最大连接数量,超过的空闲连接将被释放,如果设置为负数表示不限制(The maximum number of connections that can remain idle in the pool, without extra ones being released, or negative for no limit.)
<b>minIdle</b>	<b>0</b>	最小空闲连接:连接池中容许保持空闲状态的最小连接数量,负数表示没有现在(The maximum number of connections that can remain idle in the pool, without extra ones being released, or negative for no limit.)
<p>注意: 如果在某些负载比较大的系统中将<b>maxIdle</b>设置过小时, 很可能会出现连接关闭的同时新连接马上打开的情况.这是由于关闭连接的线程比打开的快导致的.所以, 对于这种系统中,<b>maxIdle</b>的设定值是不同的但是通常首选默认值</p>		
<p><b>(NOTE: If maxIdle is set too low on heavily loaded systems it is possible you will see connections being closed and almost immediately new connections being opened. This is a result of the active threads momentarily closing connections faster than they are opening them, causing the number of idle connections to rise above maxIdle. The best value for maxIdle for heavily loaded system will vary but the default is a good starting point.)</b></p>		
<b>maxWaitMillis</b>	<b>indefinitely</b>	最大等待时间:当没有可用连接时,连接池等待连接被归还的最大时间(以毫秒计数),超过时间则抛出异常,如果设置为-1表示无限等待(The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception, or -1 to wait indefinitely.)
<b>validationQuery</b>		<b>SQL查询</b> ,用来验证从连接池取出的连接,在将连接返回给调用者之前.如果指定,则查询必须是一个 <b>SQL SELECT</b> 并且必须返回至少一行记录(The SQL query that will be used to validate connections from this pool before returning them to the caller. If specified, this query MUST be an SQL SELECT statement that returns at least one row. If not specified, connections will be validation by calling the isValid() method.)
<b>testOnCreate</b>	<b>false</b>	指明是否在建立连接之后进行验证,如果验证失败,则尝试重新建立连接(The indication of whether objects will be validated after creation. If the object fails to validate, the

		borrow attempt that triggered the object creation will fail.)
<b>testOnBorrow</b>	<b>true</b>	指明是否在从池中取出连接前进行检验,如果检验失败,则从池中去除连接并尝试取出另一个. 注意: 设置为 <b>true</b> 后如果要生效, <b>validationQuery</b> 参数必须设置为非空字符串(The indication of whether objects will be validated before being borrowed from the pool. If the object fails to validate, it will be dropped from the pool, and we will attempt to borrow another.)
<b>testOnReturn</b>	<b>false</b>	指明是否在归还到池中前进行检验(The indication of whether objects will be validated before being returned to the pool.)
<b>testWhileIdle</b>	<b>false</b>	指明连接是否被空闲连接回收器(如果有)进行检验.如果检测失败,则连接将被从池中去除. 注意: 设置为 <b>true</b> 后如果要生效, <b>validationQuery</b> 参数必须设置为非空字符串(The indication of whether objects will be validated by the idle object evictor (if any). If an object fails to validate, it will be dropped from the pool.)
<b>timeBetweenEvictionRunsMillis</b>	<b>-1</b>	在空闲连接回收器线程运行期间休眠的时间值,以毫秒为单位.如果设置为非正数,则不运行空闲连接回收器线程(The number of milliseconds to sleep between runs of the idle object evictor thread. When non-positive, no idle object evictor thread will be run.)
<b>numTestsPerEvictionRun</b>	<b>3</b>	在每次空闲连接回收器线程(如果有)运行时检查的连接数量(The number of objects to examine during each run of the idle object evictor thread (if any).)
<b>minEvictableIdleTimeMillis</b>	<b>1000*60*30</b>	连接在池中保持空闲而不被空闲连接回收器线程(如果有)回收的最小时间值, 单位毫秒(The minimum amount of time an object may sit idle in the pool before it is eligible for eviction by the idle object evictor (if any).)
<b>softMiniEvictableIdleTimeMillis</b>	<b>-1</b>	说明(The minimum amount of time a connection may sit idle in the pool before it is eligible for eviction by the idle connection evictor, with the extra condition that at least "minIdle" connections remain in the pool. When miniEvictableIdleTimeMillis is set to a positive value, miniEvictableIdleTimeMillis is examined first by the idle connection evictor - i.e. when idle connections are visited by the evictor, idle time is first compared against miniEvictableIdleTimeMillis (without considering the number of idle connections

		in the pool) and then against softMinEvictableIdleTimeMillis, including the minIdle constraint.)
<b>maxConnLifetimeMillis</b>	<b>-1</b>	<b>说明</b> (The maximum lifetime in milliseconds of a connection. After this time is exceeded the connection will fail the next activation, passivation or validation test. A value of zero or less means the connection has an infinite lifetime.)
<b>logExpiredConnections</b>	<b>true</b>	<b>说明</b> (Flag to log a message indicating that a connection is being closed by the pool due to maxConnLifetimeMillis exceeded. Set this property to false to suppress expired connection logging that is turned on by default.
<b>connectionInitSqls</b>	<b>null</b>	<b>说明</b> (A Collection of SQL statements that will be used to initialize physical connections when they are first created. These statements are executed only once - when the configured connection factory creates the connection.)
<b>info</b>	<b>true</b>	<b>说明</b> (True means that borrowObject returns the most recently used ("last in") connection in the pool (if there are idle connections available). False means that the pool behaves as a FIFO queue - connections are taken from the idle instance pool in the order that they are returned to the pool.)
<b>poolPreparedState- ments</b>	<b>false</b>	开启池的 <b>prepared statement</b> 池功能 (Enable prepared statement pooling for this pool.)
<b>maxOpenPreparedStat e-ments</b>	<b>unlimited</b>	<b>statement池能够同时分配的打开的 statements的最大数量, 如果设置为0表示不 限制</b> (The maximum number of open statements that can be allocated from the statement pool at the same time, or negative for no limit.)
<b>NOTE</b> - Make sure your connection has some resources left for the other statements. Pooling PreparedStatements may keep their cursors open in the database, causing a connection to run out of cursors, especially if maxOpenPreparedStatements is left at the default (unlimited) and an application opens a large number of different PreparedStatements per connection. To avoid this problem, maxOpenPreparedStatements should be set to a value less than the maximum number of cursors that can be open on a Connection.		
<b>accessToUnderlyingCo nnectionAllowed</b>	<b>false</b>	控制 <b>PoolGuard</b> 是否容许获取底层连接 (Controls if the PoolGuard allows access to the underlying connection.) 默认 <b>false</b> 不开 启, 这是一个有潜在危险的功能, 不适当的编 码会造成伤害.(关闭底层连接或者在守护连接 已经关闭的情况下继续使用它).请谨慎使用,并 且仅当需要直接访问驱动的特定功能时使用.

		<p>注意: 不要关闭底层连接, 只能关闭前面的那个。 Default is false, it is a potential dangerous operation and misbehaving programs can do harmful things. (closing the underlying or continue using it when the guarded connection is already closed) Be careful and only use when you need direct access to driver specific extensions. NOTE: Do not close the underlying connection, only the original one.</p>
<b>removeAbandoned</b>	<b>false</b>	<p>标记是否删除泄露的连接,如果他们超过了<b>removeAbandonedTimeout</b>的限制.如果设置为<b>true</b>, 连接被认为是被泄露并且可以被删除, 如果空闲时间超过<b>removeAbandonedTimeout</b>. 设置为<b>true</b>可以为写法糟糕的没有关闭连接的程序修复数据库连接. (Flags to remove abandoned connections if they exceed the removeAbandonedTimeout. A connection is considered abandoned and eligible for removal if it has not been used for longer than removeAbandonedTimeout. Setting one or both of these to true can recover db connections from poorly written applications which fail to close connections.)</p>
<b>removeAbandonedTimeout</b>	<b>300</b>	<p>泄露的连接可以被删除的超时值, 单位秒 (Timeout in seconds before an abandoned connection can be removed.)</p>
<b>logAbandoned</b>	<b>false</b>	<p>标记当<b>Statement</b>或连接被泄露时是否打印程序的<b>stack traces</b>日志。被泄露的<b>Statements</b>和连接的日志添加在每个连接打开或者生成新的<b>Statement</b>,因为需要生成<b>stack trace</b>。 (Flag to log stack traces for application code which abandoned a Statement or Connection. Logging of abandoned Statements and Connections adds overhead for every Connection open or new Statement because a stack trace has to be generated.)</p>
<b>abandonedUsageTracking</b>	<b>false</b>	<p>如果为<b>true</b>, 那么连接池会记录每个方法调用时候的堆栈信息以及废弃连接的调试信息 (If true, the connection pool records a stack trace every time a method is called on a pooled connection and retains the most recent stack trace to aid debugging of abandoned connections. There is significant overhead added by setting this to true.)</p>
<p>注:如果开启"<b>removeAbandoned</b>",那么连接在被认为泄露时可能被池回收. 这个机制在 (<b>getNumIdle()</b> &lt; 2)and (<b>getNumActive()</b> &gt; <b>getMaxActive()</b> - 3)时被触发. 举例当<b>maxActive</b>=20, 活动连接为18,空闲连接为1时可以触发"<b>removeAbandoned</b>".但是活动连接只有在没有被使用的时间超过"<b>removeAbandonedTimeout</b>"时才会被删除,默认300秒.在<b>resultset</b>中游历不被计算为被使用.</p>		
<p>If you have enabled <b>removeAbandonedOnMaintenance</b> or <b>removeAbandonedOnBorrow</b> then it is possible that a connection is reclaimed by the pool because it is considered to be abandoned.</p>		



This mechanism is triggered when (getNumIdle() < 2) and (getNumActive() > getMaxTotal() - 3) and removeAbandonedOnBorrow is true; or after eviction finishes and removeAbandonedOnMaintenance is true. For example, maxTotal=20 and 18 active connections and 1 idle connection would trigger removeAbandonedOnBorrow, but only the active connections that aren't used for more then "removeAbandonedTimeout" seconds are removed (default 300 sec). Traversing a resultset doesn't count as being used. Creating a Statement, PreparedStatement or CallableStatement or using one of these to execute a query (using one of the execute methods) resets the lastUsed property of the parent connection.

## 4.2 C3P0 属性说明表

属性(Parameter)	默认值(Default)	描述(Description)
<b>user</b>		同DBCP中的username属性
<b>password</b>		同DBCP中的password属性
<b>jdbcUrl</b>		同DBCP中的jdbcUrl属性
<b>driverClass</b>		同DBCP中的driverClass属性
<b>autoCommitOnClose</b>	<b>false</b>	<p>默认值false表示回滚任何未提交的任务,设置为true则全部提交,而不是在关闭连接之前回滚</p> <p>(C3P0's default policy is to rollback any uncommitted, pending work. Setting autoCommitOnClose to true causes uncommitted pending work to be committed, rather than rolled back on Connection close.)</p> <p>*参见DBCP中的defaultAutoCommit属性</p>
<b>initialPoolSize</b>	<b>3</b>	<p>初始化连接:连接池启动时创建的初始化连接数量(The initial number of connections that are created when the pool is started.</p> <p>*参见DBCP中的initialSize属性</p>
<b>maxPoolSize</b>	<b>15</b>	<p>连接池中保留的最大连接数(Maximum number of Connections a pool will maintain at any given time.) *参见DBCP中的maxIdle属性</p>
<b>minPoolSize</b>	<b>3</b>	<p>连接池中保留的最小连接数(Minimum number of Connections a pool will maintain at any given time.) *参见DBCP中的maxIdle属性</p>
<b>maxIdleTime</b>	<b>0</b>	<p>最大等待时间:当没有可用连接时,连接池等待连接被归还的最大时间(以秒计数),超过时间则抛出异常,如果设置为0表示无限等待 (Seconds a Connection can remain pooled but unused before being discarded. Zero means idle connections never expire.) *参见DBCP中maxWaitMillis 属性</p>

<b>preferredTestQuery</b>	<b>null</b>	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一显著提高测试速度。注意：测试的表必须在初始数据源的时候就存在。(Defines the query that will be executed for all connection tests, if the default ConnectionTester (or some other implementation of QueryConnectionTester, or better yet FullQueryConnectionTester) is being used. Defining a preferredTestQuery that will execute quickly in your database may dramatically speed up Connection tests.)
<b>testConnectionOn-Checkin</b>	<b>false</b>	如果设为 <b>true</b> 那么在取得连接的同时将校验连接的有效性。(If true, an operation will be performed asynchronously at every connection checkin to verify that the connection is valid. Use in combination with idleConnectionTestPeriod for quite reliable, always asynchronous Connection testing.) *参见 <b>DBCP</b> 中的 <b>testOnBorrow</b> 属性
<b>testConnectionOn-Checkout</b>	<b>false</b>	如果设为 <b>true</b> 那么在每个 <b>connection</b> 提交的时候都将校验其有效性,但是要确保配置的 <b>preferredTestQuery</b> 的有效性(If true, an operation will be performed at every connection checkout to verify that the connection is valid. Be sure to set an efficient preferredTestQuery or automaticTestTable if you set this to true.) *参见 <b>DBCP</b> 中的 <b>testOnBorrow</b> 属性
<b>idleConnectionTest-Period</b>	<b>0</b>	如果设置大于0,表示过了多少秒检查一次空闲连接,结合 <b>testConnectionOnCheckin</b> 以及 <b>testConnectionOnCheckout</b> 使用(If this is a number greater than 0, c3p0 will test all idle, pooled but unchecked-out connections, every this number of seconds.)
<b>acquireRetryAttempts</b>	<b>30</b>	定义在从数据库获取新连接失败后重复尝试的次数, 如果小于0则表示无限制的连接。(Defines how many times c3p0 will try to acquire a new Connection from the database before giving up. If this value is less than or equal to zero, c3p0 will keep trying to fetch a Connection indefinitely.)
<b>acquireRetryDelay</b>	<b>1000</b>	两次连接中的间隔时间, 单位毫秒。(Milliseconds, time c3p0 will wait between acquire attempts.)
<b>breakAfterAcquire-Failure</b>	<b>false</b>	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 <b>getConnection()</b> 的时候继续尝试获取连接。如果为 <b>true</b> , 那么在

		<p>尝试获取连接失败后该数据源将声明已断开并永久关闭。(If true, a pooled DataSource will declare itself broken and be permanently closed if a Connection cannot be obtained from the database after making acquireRetryAttempts to acquire one. If false, failure to obtain a Connection will cause all Threads waiting for the pool to acquire a Connection to throw an Exception, but the DataSource will remain valid, and will attempt to acquire again following a call to getConnection().)</p>
checkoutTimeout	0	<p>当连接池用完时客户端调用 <b>getConnection()</b> 后等待获取新连接的时间，潮湿后将抛出 <b>SQLException</b>，如设为 <b>0</b>，则为无限期待。单位毫秒。(The number of milliseconds a client calling getConnection() will wait for a Connection to be checked-in or acquired when the pool is exhausted. Zero means wait indefinitely. Setting any positive value will cause the getConnection() call to time-out and break with an SQLException after the specified number of milliseconds.)</p>
maxStatements	0	<p>控制数据源内加载的<b>PreparedStatements</b>数量(Enable prepared statement pooling for this pool.)</p>
maxStatementsPer-Connection	0	<p>定义了连接池内单个连接所拥有的最大缓存 <b>statements</b>数(The maximum number of open statements that can be allocated from the statement pool at the same time, or negative for no limit.)</p>

### 4.3 DRUID 属性说明表

属性(Parameter)	默认值(Default)	描述(Description)
username		连接数据库的用户名
password		连接数据库的密码
jdbcUrl		同DBCP中的jdbcUrl属性
driverClassName	根据url自动识别	这一项可配可不配，如果不配置druid会根据url自动识别dbType，然后选择相应的driverClassName
initialSize	0	<p>初始化时建立物理连接的个数。初始化发生在显示调用init方法，或者第一次 <b>getConnection</b>时 *参见DBCP中的initialSize属性</p>
maxActive	8	最大连接池数量(Maximum number of

		Connections a pool will maintain at any given time.) *参见DBCP中的maxTotal属性
maxIdle	8	已经不再使用，配置了也没效果 *参见DBCP中的maxIdle属性
minIdle		最小连接池数量
maxWait		获取连接时最大等待时间，单位毫秒。配置了maxWait之后，缺省启用公平锁，并发效率会有所下降，如果需要可以通过配置useUnfairLock属性为true使用非公平锁。
poolPreparedStatements	false	是否缓存preparedStatement，也就是PSCache。PSCache对支持游标的数据库性能提升巨大，比如说oracle。
maxOpenPreparedStatements	-1	要启用PSCache，必须配置大于0，当大于0时，poolPreparedStatements自动触发修改为true。在Druid中，不会存在Oracle下PSCache占用内存过多的问题，可以把这个数值配置大一些，比如说100
testOnBorrow	true	申请连接时执行validationQuery检测连接是否有效，做了这个配置会降低性能。
testOnReturn	false	归还连接时执行validationQuery检测连接是否有效，做了这个配置会降低性能
testWhileIdle	false	建议配置为true，不影响性能，并且保证安全性。申请连接的时候检测，如果空闲时间大于timeBetweenEvictionRunsMillis，执行validationQuery检测连接是否有效。
validationQuery		用来检测连接是否有效的sql，要求是一个查询语句。如果validationQuery为null，testOnBorrow、testOnReturn、testWhileIdle都不会起作用。在mysql中通常为select 'x'，在oracle中通常为select 1 from dual
timeBetweenEvictionRunsMillis		1) Destroy线程会检测连接的间隔时间 2) testWhileIdle的判断依据
minEvictableIdleTimeMillis		Destroy线程中如果检测到当前连接的最后活跃时间和当前时间的差值大于minEvictableIdleTimeMillis，则关闭当前连接。
removeAbandoned		对于建立时间超过removeAbandonedTimeout的连接强制关闭

removeAbandoned-Timeout		指定连接建立多长时间就需要被强制关闭
logAbandoned	false	指定发生removeabandoned的时候，是否记录当前线程的堆栈信息到日志中
filters		属性类型是字符串，通过别名的方式配置扩展插件，常用的插件有： 1) 监控统计用的filter:stat 2) 日志用的filter:log4j 3) 防御sql注入的filter:wall

标签： 数据库连接池

好文要顶

关注我

收藏该文

[shaopiing](#)  
[关注 - 0](#)  
[粉丝 - 11](#)  
[+加关注](#)

14

0

» 下一篇： [FindBugs: Java 静态代码检查](#)  
posted @ 2016-03-19 11:51 shaopiing 阅读(45270) 评论(7) 编辑 收藏

评论列表

#1楼 2018-03-19 11:25 [greycat](#) [回复](#) [引用](#)

对比得很全面!

支持(0) 反对(0)

#2楼 2018-06-12 20:07 [大毅](#) [回复](#) [引用](#)

感谢分享，好文~

支持(0) 反对(0)

#3楼 2019-02-12 21:00 [曾聪聪](#) [回复](#) [引用](#)

你这个确实差不多是抄完的

支持(0) 反对(0)

#4楼 2019-05-14 16:43 [掌唇走天涯](#) [回复](#) [引用](#)

哇塞,这也太全面了吧

支持(0) 反对(0)

#5楼 2019-05-14 16:45 [掌唇走天涯](#) [回复](#) [引用](#)

您好大哥,这篇文章我能复制到我的csnd中吗

支持(0) 反对(0)

#6楼 2019-09-22 10:24 [Arebirth](#) [回复](#) [引用](#)

six

支持(0) 反对(0)

#7楼 [楼主 ] 2020-02-18 20:47 [shaopiing](#) [回复](#) [引用](#)

@ 掌唇走天涯  
当然可以，我这个也是参考各种官方文档总结的。

支持(0) 反对(0)