

原

数据库连接池 HikariCP 使用实例及配置参数详解

2018年10月29日 14:31:31

ssxueyi

阅读数 4661

更多

版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](#) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/ssxueyi/article/details/83505322>

HikariCP数据库连接池，代号：光，表示它非常快。github地址：<https://github.com/brettwooldridge/HikariCP>

jar包下载或maven引用地址：
<http://maven.outofmemory.cn/com.zaxxer/HikariCP/> 或 <https://mvnrepository.com/artifact/com.zaxxer/HikariCP>

java实例代码：

```
1 import java.sql.Connection;
2 import java.sql.ResultSet;
3 import java.sql.SQLException;
4 import java.sql.Statement;
5
6 import com.zaxxer.hikari.HikariConfig;
7 import com.zaxxer.hikari.HikariDataSource;
8
9 /**
10  * hikaricp 连接池示例
11  * @author wanghonggang
12  * 2018-10-29
13  */
14 public class HikariDemo {
15
16     public static void main(String[] args) {
17
18         //配置文件
19         HikariConfig hikariConfig = new HikariConfig();
20 //         hikariConfig.setJdbcUrl("jdbc:mysql://localhost:3306/mydata");//mysql
21         hikariConfig.setJdbcUrl("jdbc:oracle:thin:@localhost:1521:orcl");//oracle
22         hikariConfig.setDriverClassName("oracle.jdbc.driver.OracleDriver");
23         hikariConfig.setUsername("whg");
24         hikariConfig.setPassword("whg");
25         hikariConfig.addDataSourceProperty("cachePrepStmts", "true");
26         hikariConfig.addDataSourceProperty("prepStmtCacheSize", "250");
27         hikariConfig.addDataSourceProperty("prepStmtCacheSqlLimit", "2048");
28
29         HikariDataSource ds = new HikariDataSource(hikariConfig);
30         Connection conn = null;
31         Statement statement = null;
32         ResultSet rs = null;
33         try{
34
35             //创建connection
36             conn = ds.getConnection();
37             statement = conn.createStatement();
38
39             //执行sql
40             rs = statement.executeQuery("select 100 s from dual");
41
42             //取数据
43             if (rs.next()){
44                 System.out.println(rs.getString("s"));
45             }
46
47             //关闭connection
48             conn.close();
49         }
50         catch (SQLException e){
51             e.printStackTrace();
52         }
53     }
54 }
55
56 }
```

HikariCP配置说明：

参数	描述	默认值	其他
autoCommit	自动提交从池中返回的连接	true	-
connectionTimeout	等待来自池的连接的最大毫秒数	30000	如果小于250毫秒，则被重置回30秒
idleTimeout	连接允许在池中闲置的最长时间	600000	如果idleTimeout+1秒>maxLifetime 且 maxLifetime>0，则会被重置为0（代表永远不会退出）；如果idleTimeout!=0且小于10秒，则会被重置为10秒
maxLifetime	池中连接最长生命周期	1800000	如果不等于0且小于30秒则会被重置回30分钟
connectionTestQuery	如果您的驱动程序支持JDBC4，我们强烈建议您不要设置此属性	null	-
minimumIdle	池中维护的最小空闲连接数	10	minIdle<0或者minIdle>maxPoolSize,则被重置为maxPoolSize
maximumPoolSize	池中最大连接数，包括闲置和使用中的连接	10	如果maxPoolSize小于1，则会被重置。当minIdle<=0被重置为DEFAULT_POOL_SIZE则为10;如果minIdle>0则重置为minIdle的值
metricRegistry	该属性允许您指定一个 Codahale / Dropwizard MetricRegistry 的实例，供池使用以记录各种指标	null	-
healthCheckRegistry	该属性允许您指定池使用的Codahale / Dropwizard HealthCheckRegistry的实例来报告当前健康信息	null	-
poolName	连接池的用户定义名称，主要出现在日志记录和JMX管理控制台以识别池和池配置	HikariPool-1	-
initializationFailTimeout	如果池无法成功初始化连接，则此属性控制池是否将 fail fast	1	-
isolateInternalQueries	是否在其自己的事务中隔离内部池查询，例如连接活动测试	false	-
allowPoolSuspension	控制池是否可以通过JMX暂停和恢复	false	-
readOnly	从池中获取的连接是否默认处于只读模式	false	-
registerMbeans	是否注册JMX管理Bean（MBeans）	false	-
catalog	为支持 catalog 概念的数据库设置默认 catalog	null	-
connectionInitSql	该属性设置一个SQL语句，在将每个新连接创建后，将其添加到池中之前执行该语句。	null	-
driverClassName	HikariCP将尝试通过仅基于jdbcUrl的DriverManager解析驱动程序，但对于一些较旧的驱动程序，还必须指定driverClassName	null	-
transactionIsolation	控制从池返回的连接的默认事务隔离级别	null	-
validationTimeout	连接将被测试活动的最大时间量	5000	如果小于250毫秒，则会被重置回5秒
leakDetectionThreshold	记录消息之前连接可能离开池的时间量，表示可能的连接泄漏	0	如果大于0且不是单元测试，则进一步判断： (leakDetectionThreshold < SECONDS.toMillis(2) or (leakDetectionThreshold > maxLifetime && maxLifetime > 0)，会被重置为0 . 即如果要生效则必须>0，而且不能小于2秒，而且当maxLifetime > 0时不能大于maxLifetime
dataSource	这个属性允许你直接设置数据源的实例被池包装，而不是让HikariCP通过反射来构造它	null	-
schema	该属性为支持模式概念的数据库设置默认模式	null	-
threadFactory	此属性允许您设置将用于创建池使用的所有线程的java.util.concurrent.ThreadFactory的实例。	null	-
scheduledExecutor	此属性允许您设置将用于各种内部计划任务的java.util.concurrent.ScheduledExecutorService实例	null	-