原创

**HikariCP连接池及其在springboot中的配置**

2018-08-08 09:43:39 [一介那个书生](https://me.csdn.net/qq_32953079) 阅读数 21885更多

分类专栏： [Springboot](https://blog.csdn.net/qq_32953079/article/category/7493605" \t "_blank) [连接池](https://blog.csdn.net/qq_32953079/article/category/7917040)

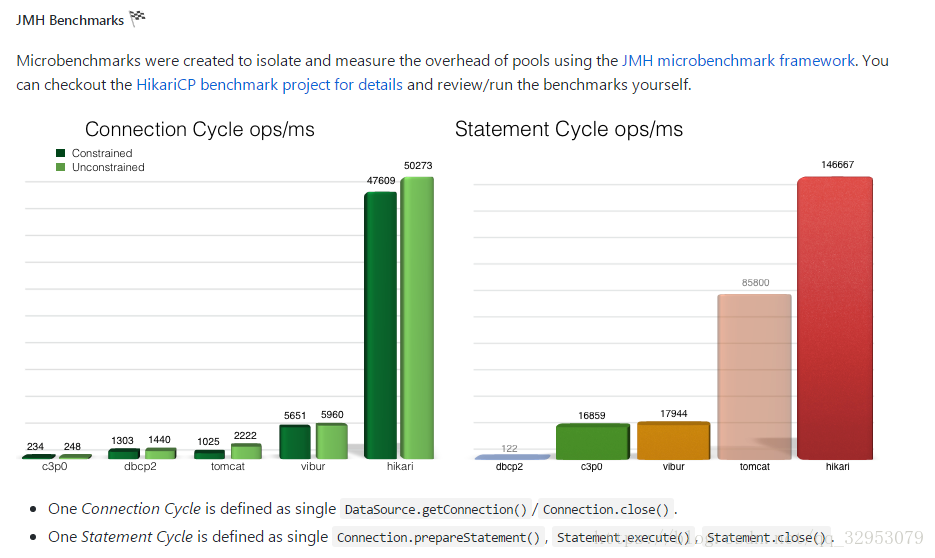
版权声明：本文为博主原创文章，遵循[CC 4.0 BY-SA](http://creativecommons.org/licenses/by-sa/4.0/)版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/qq_32953079/article/details/81502237>

        大家好，现在介绍一款非常强大，高效，并且号称“史上最快连接池”。由此可见他是有多受人喜欢，并且在springboot2.0之后，采用的默认数据库连接池就是Hikari。

         我们知道的连接池有C3P0,DBCP,它们都比较成熟稳定，但性能不是十分好。 所以有了BoneCP这个连接池，它是一个高速、免费、开源的JAVA连接池，它的性能几乎是C3P0、DBCP的25倍，十分强悍。性能对比[点击这里hikariCP,druid,tomcat-jdbc,dbcp,c3p0](https://www.tuicool.com/articles/qayayiM)

   官网也给出了一个性能对比示意图，官网在[这里](https://github.com/brettwooldridge/HikariCP)：



好，那么下面看看在springboot中如何配置Hikari连接池。主要配置如下，

| **配置项** | **描述** | **构造器默认值** | **默认配置validate之后的值** | **validate重置** |
| --- | --- | --- | --- | --- |
| autoCommit | 自动提交从池中返回的连接 | true | true | - |
| connectionTimeout | 等待来自池的连接的最大毫秒数 | SECONDS.toMillis(30) = 30000 | 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 | MINUTES.toMillis(10) = 600000 | 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 | MINUTES.toMillis(30) = 1800000 | 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 | null | - |
| minimumIdle | 池中维护的最小空闲连接数 | -1 | 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 | -1 | 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 | 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 | null | - |
| poolName | 连接池的用户定义名称，主要出现在日志记录和JMX管理控制台中以识别池和池配置 | null | 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 | 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 | false | - |
| allowPoolSuspension | 控制池是否可以通过JMX暂停和恢复 | false | false | - |
| readOnly | 从池中获取的连接是否默认处于只读模式 | false | false | - |
| registerMbeans | 是否注册JMX管理Bean（MBeans） | false | false | - |
| catalog | 为支持 catalog 概念的数据库设置默认 catalog | driver default | 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 | 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 | null | - |
| transactionIsolation | 控制从池返回的连接的默认事务隔离级别 | null | null | - |
| validationTimeout | 连接将被测试活动的最大时间量 | SECONDS.toMillis(5) = 5000 | 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 | 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 | null | - |
| schema | 该属性为支持模式概念的数据库设置默认模式 | driver default | 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 | 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 | null | - |

原版英文详细配置查看[这里](https://github.com/brettwooldridge/HikariCP)

1. ## 数据库配置
2. spring.datasource.type=com.zaxxer.hikari.HikariDataSource
3. spring.datasource.driverClassName = com.mysql.jdbc.Driver
4. spring.datasource.url = jdbc:mysql://localhost:3306/ssm?useUnicode=true&characterEncoding=utf-8&useSSL=false
5. spring.datasource.username = root
6. spring.datasource.password = root
7. ## Hikari 连接池配置 ------ 详细配置请访问：https://github.com/brettwooldridge/HikariCP
8. ## 最小空闲连接数量
9. spring.datasource.hikari.minimum-idle=5
10. ## 空闲连接存活最大时间，默认600000（10分钟）
11. spring.datasource.hikari.idle-timeout=180000
12. ## 连接池最大连接数，默认是10
13. spring.datasource.hikari.maximum-pool-size=10
14. ## 此属性控制从池返回的连接的默认自动提交行为,默认值：**true**
15. spring.datasource.hikari.auto-commit=**true**
16. ## 连接池母子
17. spring.datasource.hikari.pool-name=MyHikariCP
18. ## 此属性控制池中连接的最长生命周期，值0表示无限生命周期，默认1800000即30分钟
19. spring.datasource.hikari.max-lifetime=1800000
20. ## 数据库连接超时时间,默认30秒，即30000
21. spring.datasource.hikari.connection-timeout=30000
22. spring.datasource.hikari.connection-test-query=SELECT 1

好，启动项目，可以看到控制台

2018-08-08 09:25:01.611 INFO 8096 --- [ restartedMain] o.s.j.e.a.AnnotationMBeanExporter : Located MBean 'dataSource': registering with JMX server as MBean [com.zaxxer.hikari:name=dataSource,type=HikariDataSource]

写个查询方法，访问一下，控制台有如下信息，即表示成功应用Hikari连接池

1. 2018-08-08 09:40:00.094 INFO 8096 --- [nio-8080-exec-1] com.zaxxer.hikari.HikariDataSource : MyHikariCP - Starting...
2. 2018-08-08 09:40:00.325 INFO 8096 --- [nio-8080-exec-1] com.zaxxer.hikari.HikariDataSource : MyHikariCP - Start completed.

OKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK!!