



链滴

SpringBoot 系列 -- 整合 Druid 数据库连接池

作者: [Qiyue0726](#)

原文链接: <https://ld246.com/article/1564752370000>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



概念

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，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。（摘自[百度百科](#)）

说白了，就是可以重复使用同一个数据库连接，不再需要进行打开、关闭连接，一切都交由数据库连接去处理，一般可搭配 Mybatis 食用。

配置

导入 Maven 包

```
<properties>
  <mysql.connector.version>5.1.47</mysql.connector.version>
  <mybatis-spring-boot-starter.version>2.1.0</mybatis-spring-boot-starter.version>
  <druid-version>1.1.19</druid-version>
</properties>

<dependencies>

  <dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>${mybatis-spring-boot-starter.version}</version>
  </dependency>

  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
```

```
    <version>${mysql.connector.version}</version>
    <!--<scope>runtime</scope>-->
</dependency>

<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>druid</artifactId>
  <version>${druid-version}</version>
</dependency>

</dependencies>
```

配置数据源

这里使用的yml的配置方式，如需配置多数据源可以参考[这里](#)

```
#数据库设置
```

```
spring:
```

```
  datasource:
```

```
    driver-class-name: com.mysql.jdbc.Driver
```

```
    url: jdbc:mysql://127.0.0.1:3306/anima?characterEncoding=utf-8&autoReconnect=true&failOverReadOnly=false&useSSL=false
```

```
    username: root
```

```
    password: 3333
```

```
# 使用Druid数据源
```

```
type: com.alibaba.druid.pool.DruidDataSource
```

```
# 下面为连接池的补充设置，应用到上面所有数据源中
```

```
druid:
```

```
  # 初始化大小，最小，最大
```

```
  initialSize: 5
```

```
  minIdle: 5
```

```
  maxActive: 20
```

```
  # 配置获取连接等待超时的时间
```

```
  maxWait: 60000
```

```
  # 配置间隔多久才进行一次检测，检测需要关闭的空闲连接，单位是毫秒
```

```
  timeBetweenEvictionRunsMillis: 2000
```

```
  # 配置一个连接在池中最小生存的时间，单位是毫秒
```

```
  minEvictableIdleTimeMillis: 600000
```

```
  maxEvictableIdleTimeMillis: 900000
```

```
validationQuery: SELECT 1 FROM DUAL
```

```
testWhileIdle: true
```

```
testOnBorrow: false
```

```
testOnReturn: false
```

```
# 打开PSCache，并且指定每个连接上PSCache的大小
```

```
poolPreparedStatements: true
```

```
maxPoolPreparedStatementPerConnectionSize: 20
```

```
# 配置监控统计拦截的filters, 去掉后监控界面sql将无法统计, 'wall'用于防火墙
filters: stat, wall, log4j

# 通过connectProperties属性来打开mergeSql功能, 慢SQL记录
connectionProperties: druid.stat.mergeSql=true;druid.stat.slowSqlMillis=5000

#asyncInit是1.1.4中新增加的配置, 如果有INITIALSIZE数量较多时, 打开会加快应用启动时间
asyncInit: true
```

```
mybatis:
#实体类地址
type-aliases-package: com.sakura.anima.entity
mapper-locations: classpath:mapper/*.xml
# 开启驼峰匹配
mapUnderscoreToCamelCase: true
```

这样配置好后启动应用, 连接池就会生效了。但是这样还不能使用 Druid 的监控

配置监控

```
package com.sakura.anima.config;

import com.alibaba.druid.support.http.StatViewServlet;
import com.alibaba.druid.support.http.WebStatFilter;
import org.springframework.boot.web.servlet.FilterRegistrationBean;
import org.springframework.boot.web.servlet.ServletRegistrationBean;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

import java.util.HashMap;
import java.util.Map;

/**
 * @Author: yujianhao
 * @ClassName: druidConfig
 * @Description: 配置 Druid 监控
 * @CreateDate: 2019-07-30 09:15
 */
@Configuration
public class druidConfig {
    @Bean
    public ServletRegistrationBean druidServlet() {
        ServletRegistrationBean reg = new ServletRegistrationBean();
        reg.setServlet(new StatViewServlet());
        reg.addUrlMappings("/druid/*");
        //设置控制台管理用户
        reg.addInitParameter("loginUsername","admin");// 用户名
        reg.addInitParameter("loginPassword","3333");
        // 密码
        // 禁用HTML页面上的“Reset All”功能
        reg.addInitParameter("resetEnable","false");
        // reg.addInitParameter("allow", "127.0.0.1"); // 白名单
        // reg.addInitParameter("deny",""); // 黑名单
        return reg;
    }
}
```

```

}

@Bean
public FilterRegistrationBean filterRegistrationBean() {
    //创建过滤器
    FilterRegistrationBean filterRegistrationBean = new FilterRegistrationBean();
    filterRegistrationBean.setFilter(new WebStatFilter());
    Map<String, String> initParams = new HashMap<String, String>();
    //忽略过滤的形式
    initParams.put("exclusions", "*.js,*.gif,*.jpg,*.bmp,*.png,*.css,*.ico,/druid/*");
    filterRegistrationBean.setInitParameters(initParams);
    //设置过滤器过滤路径
    filterRegistrationBean.addUrlPatterns("/*");
    return filterRegistrationBean;
}
}

```

最后在浏览器输入<http://localhost:8080/druid>就可以使用 Druid 的监控了，用户名、密码就是上面设置的那个。

测试

按照上面的配置虽然已经可以生效了，但具体有没有效果我们还不知道，接下来就来试试看性能如何。

查询服务

首先写一个简单查询数据库的服务

```

@Override
public void testDB() {

    int num = 0;

    while (num < 1000){

        System.out.println("第 " + num + " 次 ==> " + animalInfoMapper.selectByPrimaryKey(
1));
        num++;
    }
}
}

```

测试用例

```

package com.sakura.anima.service;

import com.sakura.anima.AnimaApplication;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringRunner;

```

```

import java.util.Date;

/**
 * @Author: yujianhao
 * @ClassName: getInfoServiceTest
 * @Description: 测试查询数据库用时
 * @CreateDate: 2019-07-29 15:55
 */
//需要注意：如果我们使用的是JUnit 4，那么需要添加@RunWith(SpringRunner.class)否则所有注
//将会被忽略。
//如果你使用的是JUnit5,那么在 SpringBootTest 上没有必要添加 @ExtendWith,因为@...Test已经
//加了ExtendWith
@RunWith(SpringRunner.class)
@SpringBootTest(classes={AnimaApplication.class})
public class getInfoServiceTest {

    @Autowired
    getInfoService getInfoService;

    @Test
    public void testDB() {

        Date date1 = new Date();

        getInfoService.testDB();

        Date date2 = new Date();

        System.out.println("用时: " + (date2.getTime() - date1.getTime()) + " ms");
    }
}

```

开始测试

由于 **Mybatis** 是自带连接池，所以我们就先来看看自带的连接池性能如何。

当然`application.yml`也得做相应的修改

#数据库设置

spring:

```

datasource:
  driver-class-name: com.mysql.jdbc.Driver
  url: jdbc:mysql://127.0.0.1:3306/anima?characterEncoding=utf-8&autoReconnect=true&failOverReadOnly=false&useSSL=false
  username: root
  password: 3333

```

mybatis:

```

#实体类地址
type-aliases-package: com.sakura.anima.entity

```

```
mapper-locations: classpath:mapper/*.xml
# 开启驼峰匹配
mapUnderscoreToCamelCase: true
```

```
第 994 次 ==> com.sakura.anima.entity.animaInfo@7a9eccc4
第 995 次 ==> com.sakura.anima.entity.animaInfo@5f05c8d6
第 996 次 ==> com.sakura.anima.entity.animaInfo@64942607
第 997 次 ==> com.sakura.anima.entity.animaInfo@7cfac39f
第 998 次 ==> com.sakura.anima.entity.animaInfo@50eb4a2c
第 999 次 ==> com.sakura.anima.entity.animaInfo@3069a360
用时: 1736 ms
```

测试五次的最快时间是1736ms

再来看看**不使用连接池**的性能

#数据库设置

spring:

datasource:

driver-class-name: com.mysql.jdbc.Driver

url: jdbc:mysql://127.0.0.1:3306/anima?characterEncoding=utf-8&autoReconnect=true&failOverReadOnly=false&useSSL=false

username: root

password: 3333

不使用连接池

type: com.mysql.jdbc.jdbc2.optional.MysqlConnectionPoolDataSource

mybatis:

#实体类地址

type-aliases-package: com.sakura.anima.entity

mapper-locations: classpath:mapper/*.xml

开启驼峰匹配

mapUnderscoreToCamelCase: true

```
第 995 次 ==> com.sakura.anima.entity.animaInfo@2bb62414
第 996 次 ==> com.sakura.anima.entity.animaInfo@184823ed
第 997 次 ==> com.sakura.anima.entity.animaInfo@46c10083
第 998 次 ==> com.sakura.anima.entity.animaInfo@37c5284a
第 999 次 ==> com.sakura.anima.entity.animaInfo@10f477e2
用时: 5268 ms
```

测试五次的最快时间是5268ms，大部分时间应该是花费在了连接、关闭数据库上。

最后来看看使用 **Druid** 的性能如何

这里 yml 不需要改动，只需和刚才一开始配置的一样就行

```
第 994 次 ==> com.sakura.anima.entity.animaInfo@76544c0a
第 995 次 ==> com.sakura.anima.entity.animaInfo@650c405c
第 996 次 ==> com.sakura.anima.entity.animaInfo@748ac6f3
第 997 次 ==> com.sakura.anima.entity.animaInfo@6ffff46bf
第 998 次 ==> com.sakura.anima.entity.animaInfo@3aaa3c39
第 999 次 ==> com.sakura.anima.entity.animaInfo@70f68288
用时: 1511 ms
```

测试五次的最快时间1511ms, 还是比 Mybatis 自带的连接池快上那么一丢丢。阿里的东西就是好用