



链滴

Springboot 基于 druid 的多数据源配置

作者: [michael](#)

原文链接: <https://ld246.com/article/1672218278831>

来源网站: [链滴](#)

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

序

经常在开发过程中遇到多数据源操作的需求，这里记录一下springboot的非分布式事务操作多个数据的配置，即这多个数据源是在不同的事务中去操作的。

主数据源配置

我们这里使用的不同的注解来划分不同的数据源。主数据源依然扫描的是mybatis自带的Mapper注

```
import javax.sql.DataSource;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.session.SqlSessionFactory;
import org.mybatis.spring.SqlSessionFactoryBean;
import org.mybatis.spring.SqlSessionTemplate;
import org.mybatis.spring.annotation.MapperScan;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Primary;
import org.springframework.jdbc.datasource.DataSourceTransactionManager;
import com.alibaba.druid.spring.boot.autoconfigure.DruidDataSourceBuilder;

/**
 * 数据源注入配置
 *
 * @ClassName: DataSourceConfig
 * @author: Michael
 * @date: 2021-2-28 17:17:44
 * @since: v1.0
 */
@Configuration
@MapperScan(basePackages = { "io.github.junxworks.ep.*", "com.yrsc.bp.*" }, annotationClass = Mapper.class, sqlSessionTemplateRef = "mainSqlSessionTemplate")
public class DataSourceConfig {

    @Bean
    @Primary
    @ConfigurationProperties("spring.datasource.druid.main")
    public DataSource mainDataSource() {
        return DruidDataSourceBuilder.create().build();
    }

    @Primary
    @Bean(name = "mainSqlSessionFactory")
    public SqlSessionFactory mainSqlSessionFactory(@Qualifier("mainDataSource") DataSource
dataSource) throws Exception {
        SqlSessionFactoryBean bean = new SqlSessionFactoryBean();
        bean.setDataSource(dataSource);
        SqlSessionFactory factory = bean.getObject();
        org.apache.ibatis.session.Configuration config = factory.getConfiguration();
    }
}
```

```

        config.setMapUnderscoreToCamelCase(true);
        config.setLazyLoadingEnabled(true);
        config.setDefaultStatementTimeout(300);
        return factory;
    }

    @Primary
    @Bean(name = "mainTransactionManager")
    public DataSourceTransactionManager mainTransactionManager(@Qualifier("mainDataSource") DataSource dataSource) {
        return new DataSourceTransactionManager(dataSource);
    }

    @Primary
    @Bean(name = "mainSqlSessionFactory")
    public SqlSessionFactory mainSqlSessionFactory(@Qualifier("mainSqlSessionFactory") SqlSessionFactory sqlSessionFactory) throws Exception {
        return new SqlSessionFactory(sqlSessionFactory);
    }
}

```

次数据源配置

次数据源使用的注解是自定义的LM注解，只要mapper接口上添加了LM注解，则走LM数据源。

```

import java.lang.annotation.Documented;
import java.lang.annotation.ElementType;
import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.annotation.Target;
import javax.sql.DataSource;
import org.apache.ibatis.session.SqlSessionFactory;
import org.mybatis.spring.SqlSessionFactoryBean;
import org.mybatis.spring.SqlSessionTemplate;
import org.mybatis.spring.annotation.MapperScan;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.boot.context.properties.ConfigurationProperties;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.jdbc.datasource.DataSourceTransactionManager;
import com.alibaba.druid.spring.boot.autoconfigure.DruidDataSourceBuilder;
import com.yrsc.bp.config.LmDataSourceConfig.LM;

@Configuration
@MapperScan(basePackages = { "com.yrsc.bp.*" }, annotationClass = LM.class, sqlSessionTemplateRef = "lmSqlSessionTemplate")
public class LmDataSourceConfig {

    @Bean(name = "lmDataSource")
    @ConfigurationProperties("spring.datasource.druid.lm")
    public DataSource investDataSource() {
        return DruidDataSourceBuilder.create().build();
    }
}

```

```

}

@Bean(name = "ImSqlSessionFactory")
public SqlSessionFactory ImSqlSessionFactory(@Qualifier("ImDataSource") DataSource dat
Source) throws Exception {
    SqlSessionFactoryBean bean = new SqlSessionFactoryBean();
    bean.setDataSource(dataSource);
    SqlSessionFactory factory = bean.getObject();
    org.apache.ibatis.session.Configuration config = factory.getConfiguration();
    config.setMapUnderscoreToCamelCase(true);
    config.setLazyLoadingEnabled(true);
    config.setDefaultStatementTimeout(300);
    return factory;
}

@Bean(name = "ImTransactionManager")
public DataSourceTransactionManager ImTransactionManager(@Qualifier("ImDataSource")
DataSource dataSource) {
    return new DataSourceTransactionManager(dataSource);
}

@Bean(name = "ImSqlSessionTemplate")
public SqlSessionTemplate ImSqlSessionTemplate(@Qualifier("ImSqlSessionFactory") SqlSe
sionFactory sqlSessionFactory) throws Exception {
    return new SqlSessionTemplate(sqlSessionFactory);
}

@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
@Documented
public static @interface LM {
}
}

```

Application配置

```

spring:
  datasource:
    type: com.alibaba.druid.pool.DruidDataSource
    druid:
      main: #数据服务数据源
        driverClassName: com.mysql.cj.jdbc.Driver
        url: jdbc:mysql://dburl/test?serverTimezone=Asia/Shanghai&allowMultiQueries=true&u
eUnicode=true&characterEncoding=UTF-8&autoReconnect=true&rewriteBatchedStatement
=true&zeroDateTimeBehavior=convertToNull
        username: test
        password: "test"
        initial-size: 1
        max-active: 100
        min-idle: 1
        max-wait: 60000
        pool-prepared-statements: true
        max-pool-prepared-statement-per-connection-size: 20

```

```
time-between-eviction-runs-millis: 60000
min-evictable-idle-time-millis: 300000
validation-query: SELECT 1
test-while-idle: true
test-on-borrow: false
test-on-return: false
lm: #数据服务数据源
  driverClassName: com.mysql.cj.jdbc.Driver
  url: jdbc:mysql://dburl/lmtest?serverTimezone=Asia/Shanghai&allowMultiQueries=true
  useUnicode=true&characterEncoding=UTF-8&autoReconnect=true&rewriteBatchedStatements=true&zeroDateTimeBehavior=convertToNull
  username: lmtest
  password: "lmtest"
  initial-size: 1
  max-active: 100
  min-idle: 1
  max-wait: 60000
  pool-prepared-statements: true
  max-pool-prepared-statement-per-connection-size: 20
  time-between-eviction-runs-millis: 60000
  min-evictable-idle-time-millis: 300000
  validation-query: SELECT 1
  test-while-idle: true
  test-on-borrow: false
  test-on-return: false
#下一个数据源
#公共参数
stat-view-servlet:
  enabled: true
  url-pattern: /druid/*
  #login-username: admin
  #login-password: admin
filter:
  stat:
    log-slow-sql: true
    slow-sql-millis: 1000
    merge-sql: true
  wall:
    config:
      multi-statement-allow: true
```