Atitit sprbt 多数据源mltds datasource multi

目录

[第一节 App cfg 1](#_Toc31428)

[第二节 Cfg bean 1](#_Toc11779)

[第三节 Use 4](#_Toc7030)

## App cfg

spring.datasource.db2.jdbc-url=jdbc:sqlite:db2.db

spring.datasource.db2.driver-class-name=org.sqlite.JDBC

Note,,cant only smpl db2.url...must db2.jdbc-url=

## Cfg bean

package sprbtPKg;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

import javax.sql.DataSource;

import org.apache.ibatis.session.SqlSessionFactory;

import org.apache.log4j.Logger;

import org.mybatis.spring.SqlSessionFactoryBean;

import org.mybatis.spring.SqlSessionTemplate;

import org.mybatis.spring.annotation.MapperScan;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.annotation.Qualifier;

/\*

\* \*/

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.builder.SpringApplicationBuilder;

import org.springframework.boot.context.properties.ConfigurationProperties;

import org.springframework.boot.jdbc.DataSourceBuilder;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

import org.springframework.core.io.Resource;

import org.springframework.core.io.support.PathMatchingResourcePatternResolver;

import other.Log4jTet;

import util.MybatisUtilV55;

@Configuration

@ComponentScan(lazyInit = true)

@SpringBootApplication

public class Application {

public static ConfigurableApplicationContext cfgAppContext;

public static void main(String[] args) throws Exception {

Logger logger = Logger.getLogger(Log4jTet.class);

logger.info("start ...");

cfgAppContext = SpringApplication.run(Application.class, args);

}

@Bean(name = "ds2")

@ConfigurationProperties(prefix = "spring.datasource.db2")

public DataSource newhomeDbDataSource() {

return DataSourceBuilder.create().build();

}

@Bean(name = "sqlSessionFactoryDb2")

public SqlSessionFactory sqlSessionFactoryDb2(@Qualifier("ds2") DataSource dataSourceDb2) throws Exception {

SqlSessionFactoryBean SqlSessionFactoryBean1 = new SqlSessionFactoryBean();

SqlSessionFactoryBean1.setDataSource(dataSourceDb2);

// , mapper/\*\*/\*xMap.Mxml

Resource[] ResourceMappers = MybatisUtilV55.getRsces("classpath:mapper/\*\*/\*.xml,classpath:mapper/\*\*/\*Map.xml");

SqlSessionFactoryBean1.setMapperLocations(ResourceMappers);

return SqlSessionFactoryBean1.getObject();

}

//

// @Bean

// public SqlSessionTemplate sqlSessionTemplateDb2() throws Exception {

// return new SqlSessionTemplate(sqlSessionFactoryDb2());

// }

//

// @Autowired

// @Qualifier("ds2")

// private DataSource dataSourceDb2;

}

**public** **static** Resource[] getRsces(String mapPath) **throws** IOException {

Set<Resource> resourcesSet = **new** HashSet();

String[] paths = mapPath.split(",");

**for** (String pth : paths) {

Resource[] resources = **new** PathMatchingResourcePatternResolver().getResources(pth);

resourcesSet.addAll(Arrays.*asList*(resources));

}

// [file

// [C:\Users\ATI\eclipse-workspace\miniCodePrj\target\classes\mapper\rmDao.xml]

Resource[] ResourceMappers = resourcesSet.stream().toArray(Resource[]::**new**);

**return** ResourceMappers;

}

## Use

**package** sprbtPKg;

**import** org.apache.ibatis.session.SqlSession;

**import** org.apache.ibatis.session.SqlSessionFactory;

**import** org.apache.log4j.Logger;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** org.springframework.web.bind.annotation.RestController;

**import** other.Log4jTet;

@RestController

**public** **class** HelloController {

@Autowired

SqlSession SqlSession1;

@RequestMapping("/db")

Object db() {

Logger logger = Logger.*getLogger*(Log4jTet.**class**);

logger.info( SqlSession1.selectList("qry1","select 1 t1"));

// org.sqlite.JDBC

// return SqlSession1.selectList("qry1","select 2 t22");

SqlSessionFactory SqlSessionFactory2= (SqlSessionFactory) Application.*cfgAppContext*.getBean("sqlSessionFactoryDb2");

SqlSession SqlSession2=SqlSessionFactory2.openSession(**true**);

**return** SqlSession2.selectList("qry1","select 3 t33");

}