Atitit 命令行执行springboot程序

目录

[1.1. 执行spel表达式，调用app main，获取context 1](#_Toc8669)

[1.2. 直接在Application main函数内执行 1](#_Toc8491)

[1.3. CommandLineRunner 1](#_Toc21361)

[2. 执行顺序 [{frm CommandLineRunnerImp =frm CommandLineRunnerImp }] 3](#_Toc9599)

## 执行spel表达式，调用app main，获取context

## 直接在Application main函数内执行

**public** **class** tApplication

{

**public** **static** **void** main(String[] args) {

ConfigurableApplicationContext context=SpringApplication.*run*( tApplication.**class**, args);

MybatisMapper MybatisMapper1 = context.getBean(MybatisMapper.**class**);

System.***out***.println(MybatisMapper1.querySql("select 'frm appmain'"));

}

}

## CommandLineRunner

在spring boot应用中，我们可以在程序启动之前执行任何任务。为了达到这个目的，我们需要使用CommandLineRunner或ApplicationRunner接口创建bean，spring boot会自动监测到它们。这两个接口都有一个run()方法，在实现接口时需要覆盖该方法，并使用@Component注解使其成为bean。CommandLineRunner和ApplicationRunner的作用是相同的。不同之处在于CommandLineRunner接口的run()方法接收String数组作为参数，而ApplicationRunner接口的run()方法接收ApplicationArguments对象作为参数。当程序启动时，我们传给main()方法的参数可以被实现CommandLineRunner和ApplicationRunner接口的类的run()方法访问。我们可以创建多个实现CommandLineRunner和ApplicationRunner接口的类。为了使他们按一定顺序执行，可以使用@Order注解或实现Ordered接口。

CommandLineRunner和ApplicationRunner接口的run()方法在SpringApplication完成启动时执行。启动完成之后，应用开始运行。CommandLineRunner和ApplicationRunner的作用是在程序开始运行前执行任务或记录信息。

import org.apache.ibatis.session.SqlSession;

import org.apache.ibatis.session.SqlSessionFactory;

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

import org.springframework.boot.ApplicationArguments;

import org.springframework.boot.ApplicationRunner;

import org.springframework.boot.CommandLineRunner;

import org.springframework.stereotype.Component;

import com.kok.sport.utils.MybatisMapper;

@Component

public class CommandLineRunnerImp implements CommandLineRunner {

public static void main(String[] args) {

// SpringbootRunner

// TopLaunchApplication.run(CommonConstant.APPLICATION\_SERVICE\_FS\_BOOK\_NAME,

// TopFsApplication.class, args);

}

// @Autowired

// private IResourceFeginClient resourceFeginClient;

@Autowired

SqlSessionFactory sqlSessionFactory;

@Override

public void run(String... args) throws Exception {

SqlSession session = sqlSessionFactory.openSession(true);

MybatisMapper MybatisMapper1 = session.getMapper(MybatisMapper.class);

System.out.println(MybatisMapper1.querySql("select 'frm CommandLineRunnerImp '"));

System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_=================\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*atti::");

// System.exit(0);

//

}

}

# 执行顺序 [{frm CommandLineRunnerImp =frm CommandLineRunnerImp }]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_=================\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*atti::

[{frm appmain=frm appmain}]