

1.Spring Aware

参见git ch3 aware

2.多线程

Spring通过任务执行器(TaskExecutor)来实现多线程和并发编程. 使用ThreadPoolTaskExecutor可实现一个基于线程池的TaskExecutor.

实际开发中任务一般是非阻碍的, 即异步的, 所以要在配置类通过@EnableAsync开启对异步任务的支持, 并通过在实际执行的Bean的方法中使用@Async注解来声明其是一个异步任务.

1. 配置类需要实现AsyncConfigurer接口并重写getAsyncExecutor方法, 并返回一个ThreadPoolTaskExecutor, 这样我们就得到一个基于线程池TaskExecutor

2. 通过@Async注解表明该方法是个异步方法, 如果注解在类级别, 则表明该类所有方法都是异步方法, 这里的方法自动被注入使用ThreadPoolTaskExecutor作为TaskExecutor.

3.计划任务

@EnableScheduling来开启对计划任务的支持, 然后在要执行任务的方法上注解@Scheduled, 声明这是一个计划任务.

spring通过@Scheduled支持多种类型的计划任务, 包含cron, fixDelay, fixRate等

fixedRate 固定时间执行, 与任务执行时间长短无关, 每隔n秒执行

cron是Linux系统下的定时任务

4.条件注解@Conditional

实现接口Condition, 重写matches方法

5.组合注解与元注解

元注解就是可以注解到别的注解上的注解, 被注解的注解称之为组合注解, 组合注解具备注解其上的元注解的功能.

6.@Enable*注解工作原理

@EnableAspectJAutoProxy开启对AspectJ自动代理的支持

@EnableAsync开启对异步方法的支持

@EnableScheduling开启计划任务的支持

@EnableMvc开启Web MVC的配置支持

@EnableConfigurationProperties开启对@ConfigurationProperties注解配置Bean的支持

@EnableJpaRepositories开启对Spring Data JPA Repository的支持

@EnableTransactionManagement开启注解式事务的支持

@EnableCaching开启注解式缓存支持

7.测试

@ContextConfiguration(classes) 加载配置类

@WebAppConfiguration("src/main/resources") 指定web资源位置