1、@PostConstruct //初始化方法的注解方式  等同与init-method=init

2、@PreDestroy //销毁方法的注解方式  等同于destory-method=destory222

3、[cron定时任务表达式图解](#cron定时任务)

通过@EnableScheduling开启定时任务

总共有：秒、分、时、日、月、周（星期）、年，7个组成部分，springboot不用年，所以cron表达式有6个部分。每个部分规则相同，以下分析第一部分即可。

1. 每秒 允许的通配符[, - \* /]： 如果指定这一栏 显示 0
2. 周期从\_\_到\_\_： 如果指定这一栏 显示 n-m，间隔为1秒，即从n到m秒运行
3. 从 \_\_秒开始,每\_\_秒执行1次： 如果指定这一栏 显示 n/m，从n秒开始，每m秒运行一次
4. 指定： 如果指定这一栏 显示 a,b,c,d，即指定那几秒运行

例：

0/2 0/3 \* \* \* ?：表示没3分钟运行开始，从0秒间隔2秒运行一次

1 44 11 \* \* ？：表示每天的11：44:01秒执行一次

4、异步任务

通过@EnableAsync打开异步任务，和定时任务不冲突

异步任务是被调用的，而不是像定时任务一样，自动执行，当然，也可以通过定时任务调用异步任务，来进行异步任务定时执行。

使用场景：

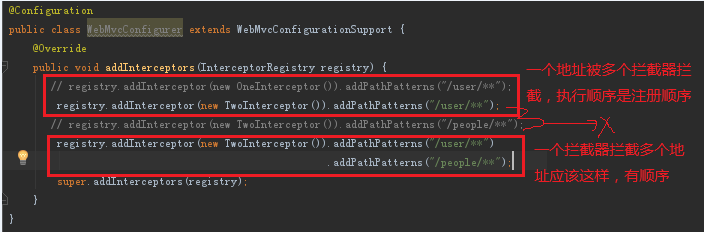
1）发送短信 2）发送邮件

3）App消息推送 4）节省运维凌晨发布任务时间提供效率

5、拦截器

与sprigmvc过滤器完全相同

1. 使用注解@Configuration配置拦截器
2. 继承WebMvcConfigurerAdapter，2.0后继承WebMvcConfigurationSupport
3. 重写addInterceptors添加需要的拦截器地址，地址不加项目名



6、日志与日志分析可以用mongodb做数据库

7、热部署

场景：

1）本地调试

2）线上发布

一些网站或服务，全年不间断运行，即使重新发布程序后也不需要重启服务！这个功能很 关键！

优点：

1. 无论本地还是线上，都适用
2. 无需重启服务器，提高开发、调试效率；提高发布、运维效率，降低运维成本

热部署与热加载的联系

1. 不重启服务器编译/部署项目
2. 基于Java的类加载器实现

热部署与热加载的区别

1. 部署方式：热部署在服务器运行时重新部署项目；热加载在运行时重新加载更改的类
2. 实现原理：热部署直接重新加载整个应用；热加载在运行时重新加载class（后台线程去不停的检查）
3. 使用场景：热部署更多的是在生产环境使用；热加载更多的是在开发环境中使用；热加载需要频繁的检查所以生产不需要，也不好

Java类的热部署实现

1. 类的热加载，可以通过配置热加载，来配置热部署
2. 配置Tomcat
3. 直接把项目放在webapps里，webapps里的内容改变了，会直接重新部署，因为有autoDeploy



1. 在server.xml中配置<context/>标签，

将web项目复制到docBase文件夹中，大的项目文件夹不复制，只复制内容，用hot作为查询的项目名



1. 在conf\catalina\localhost中加一个xml文件，文件名为项目名



Java类的热加载：

热加载最关键的是两个属性：

1）一个hashMap记录所有热加载的类，如果集合了没有，说明还没有经过自定义的类加载器，调用自定义类加载器进行加载，设置最近修改时间！

2）最后修改时间：文件都有一个最后修改时间，对比hashMap中的类和文件类的最后修改时间，如果不一致，则重新热加载

热部署实现：

1. 使用Spring Loaded
2. 添加依赖，用maven命令运行程序
3. Run as – java application
4. 使用Spring-boot-devtools

添加依赖 spring-boot-devtools

8、springboot项目中的resource/static/文件是放置静态文件的，比如，css、js文件;resource/templates/文件是放置web项目里的文件

1. cron定时任务

