1、安装包：Volo.Abp.BackgroundJobs.Quartz

2、添加Module依赖

[DependsOn(

    //...other dependencies

    typeof(AbpBackgroundJobsQuartzModule)

)]

public class YourModule : AbpModule

{

}

3、新建作业任务

    public class InventoryAlertJob : IJob, ITransientDependency

    {

        protected IServiceProvider Provider { get; }

        public InventoryAlertJob(IServiceProvider provider)

        {

            this.Provider = provider;

        }

        public async Task Execute(IJobExecutionContext context)

        {

            ...

        }

    }

4、配置作业任务

public override void ConfigureServices(

    ServiceConfigurationContext context)

{

    // 添加一个调度器

    context.Services.AddQuartz(q =>

    {

        // 调度器ID和Name

        q.SchedulerId = "InventoryAlertScheduler";

        q.SchedulerName = "InventoryAlertScheduler";

        // 关联到一个触发器和一个任务

        q.ScheduleJob<InventoryAlertJob>(

            // 关联触发器

            trigger => trigger

                    // 触发器的标识为 InventoryAlertTrigger

                    .WithIdentity("InventoryAlertTrigger")

                    // 触发器关联的任务标识为 InventoryAlertJob

                    .ForJob("InventoryAlertJob")

                    // 启动后立即执行一次任务

                    .StartNow()

                    // 每隔

                    .WithSimpleSchedule(x =>

                        // 每隔20秒触发一次

                        x.WithIntervalInSeconds(20)

                        // 指定触发器将无限重复下去

                        .RepeatForever()

                    ),

            // 关联任务

            job => job.WithIdentity("InventoryAlertJob")

        );

    });

}

1. 时间配置

使用WithCronSchedule可以更好的配置触发时间

                trigger => trigger

                        .WithIdentity("InvoicingTrigger")

                        .ForJob("InvoicingJob")

                         // 每月最后一天16点触发

                        .WithCronSchedule("0 0 16 L \* ?"),

常用示例:

0 0 12 \* \* ? 每天12点触发

0 15 10 ? \* \* 每天10点15分触发

0 15 10 \* \* ? 每天10点15分触发

0 15 10 \* \* ? \* 每天10点15分触发

0 15 10 \* \* ? 2005 2005年每天10点15分触发

0 \* 14 \* \* ? 每天下午的 2点到2点59分每分触发

0 0/5 14 \* \* ? 每天下午的 2点到2点59分(整点开始，每隔5分触发)

0 0/5 14,18 \* \* ? 每天下午的 2点到2点59分(整点开始，每隔5分触发)

每天下午的 18点到18点59分(整点开始，每隔5分触发)

0 0-5 14 \* \* ? 每天下午的 2点到2点05分每分触发

0 10,44 14 ? 3 WED 3月分每周三下午的 2点10分和2点44分触发 （特殊情况，在一个时间设置里，执行两次或 两次以上的情况）

0 59 2 ? \* FRI 每周5凌晨2点59分触发；

0 15 10 ? \* MON-FRI 从周一到周五每天上午的10点15分触发

0 15 10 15 \* ? 每月15号上午10点15分触发

0 15 10 L \* ? 每月最后一天的10点15分触发

0 15 10 ? \* 6L 每月最后一周的星期五的10点15分触发

0 15 10 ? \* 6L 2002-2005 从2002年到2005年每月最后一周的星期五的10点15分触发

0 15 10 ? \* 6#3 每月的第三周的星期五开始触发

0 0 12 1/5 \* ? 每月的第一个中午开始每隔5天触发一次

0 11 11 11 11 ? 每年的11月11号 11点11分触发(光棍节)

1. 指定使用UTC时间触发

.WithCronSchedule("0 0 0 1 \* ?", cron => { cron.InTimeZone(TimeZoneInfo.Utc); })

如上UTC 每月1号0:0:0 触发

7、触发器详细配置

https://www.quartz-scheduler.net/documentation/quartz-3.x/tutorial/crontriggers.html#example-cron-expressions