

مقدمه

اگر قصد اجرای برخی کارها به صورت زمانبندی شده و در فواصل زمانی مشخص را دارید، این مقاله به شما کمک خواهد کرد تا به بهترین شکل ممکن آن را انجام دهید. کارهایی مانند ارسال خبرنامه، فرستادن SMS تبریک تولد یا هماهنگ سازی داده‌ها بین دو منبع داده از جمله اعمالی هستند که باید به صورت زمانبندی شده انجام شوند.

کتابخانه‌ی Quartz.NET، از کتابخانه‌ای با نام Quartz و از زبان Java به NET. منتقل شده است. Quartz.NET، رایگان و باز متن است و از طریق آدرس <http://quartznet.sourceforge.net> در دسترس است. از طریق NuGet نیز می‌توانید با تایپ عبارت quartz در فرم مربوطه، این کتابخانه را نصب کنید. این کتابخانه را در برنامه‌های Web و Desktop (حتی یک Shared Server) تست کردم و به خوبی انجام وظیفه می‌کند.

شروع کار با Quartz.NET

ضمن در اختیار قرار دادن امکانات فوق العاده و انعطاف پذیری بسیار، کار با این کتابخانه آسان و از فرایندی منطقی تبعیت می‌کند. فرایند اجرای یک روال زمانبندی شده از طریق Quartz.NET، از چهار مرحله‌ی اصلی تشکیل شده است.

(1) پیاده سازی اینترفیس IJob

(2) مشخص کردن جزئیات روال با اینترفیس IJobDetail

(3) مشخص کردن تنظیمات زمان با استفاده از اینترفیس ITrigger

(4) مدیریت اجرا با استفاده از اینترفیس IScheduler

مثالی را بررسی می‌کنیم. در این مثال قصد داریم تا عبارتی را همراه با تاریخ و زمان جاری در یک فایل ذخیره کنیم. این پیغام باید 3 بار و در فواصل زمانی 10 ثانیه به فایل اضافه شود. در پایان، فایلی خواهیم داشت که در سه خط، یک عبارت، همراه با تاریخ و زمان‌های مختلف را که 10 ثانیه با یکدیگر اختلاف دارند در خود ذخیره کرده است. ابتدا کار زمانبندی شده را با ارائه‌ی پیاده سازی برای متد Execute اینترفیس IJob این کتابخانه ایجاد می‌کنیم. وارد کردن فضای نام Quartz را فراموش نکنید.

```
namespace SchedulerDemo.Jobs
{
    using System;
    using System.IO;
    using Quartz;

    public class HelloJob : IJob
    {
        public void Execute(IJobExecutionContext context)
        {
            // for web apps
            // string path = System.Web.Hosting.HostingEnvironment.MapPath("~/Data/Log.txt");

            // for desktop apps
            string path = @"C:\Log.txt";

            using (StreamWriter sw = new StreamWriter(path, true))
            {
                sw.WriteLine("Message from HelloJob " + DateTime.Now.ToString());
            }
        }
    }
}
```

در اینترفیس IJob در ASP.NET، به شیء HttpContext دسترسی ندارید، بنابراین در صورتی که قصد داشته باشید از متدی مانند Server.MapPath استفاده کنید، توفیقی به دست نخواهید آورد. در عوض می‌توانید از متد System.Web.Hosting.HostingEnvironment.MapPath استفاده کنید.

حال، زمان انجام تنظیمات مختلف برای اجرای روال مربوطه است. بهتر است تا interfaceی ایجاد و متدی با نام Run در آن

داشته باشیم.

```
namespace SchedulerDemo.Interfaces
{
    public interface ISchedule
    {
        void Run();
    }
}
```

حال، پیاده سازی خود را برای این interface ارائه می‌دهیم.

```
namespace SchedulerDemo.Jobs
{
    using System;
    using Quartz;
    using Quartz.Impl;
    using SchedulerDemo.Interfaces;
    using SchedulerDemo.Jobs;

    public class HelloSchedule : ISchedule
    {
        public void Run()
        {
            //DateTimeOffset startTime = DateBuilder.NextGivenSecondDate(null, 2);
            DateTimeOffset startTime = DateBuilder.FutureDate(2, IntervalUnit.Second);

            IJobDetail job = JobBuilder.Create<HelloJob>()
                .WithIdentity("job1")
                .Build();

            ITrigger trigger = TriggerBuilder.Create()
                .WithIdentity("trigger1")
                .StartAt(startTime)
                .WithSimpleSchedule(x =>
                    x.WithIntervalInSeconds(10).WithRepeatCount(2))
                .Build();

            ISchedulerFactory sf = new StdSchedulerFactory();
            IScheduler sc = sf.GetScheduler();
            sc.ScheduleJob(job, trigger);

            sc.Start();
        }
    }
}
```

معرفی فضاهای نام Quartz و Quartz.Impl را فراموش نکنید.

از حالا، به روالی که قرار است به صورت زمانبندی شده اجرا شود، "وظیفه" می‌گوییم. ابتدا باید مشخص کنیم که وظیفه در چه زمانی پس از اجرای برنامه شروع به اجرا کند. از آنجا که پایه و اساس زمانبندی، بر تاریخ و ساعت استوار است، کتابخانه‌ی Quartz.NET، روش‌ها و امکانات بسیاری را برای تعیین زمان در اختیار قرار می‌دهد. با بررسی تمامی آنها، ساده‌ترین و منعطف‌ترین را به شما معرفی می‌کنم. کلاس DateBuilder که همراه با Quartz.NET وجود دارد، امکان تعیین زمان را به اشکال مختلف می‌دهد. در خط 14، از متد FutureDate این کلاس استفاده شده است که خوانایی بهتری نسبت به بقیه‌ی متدها دارد. پارامتر اول این متد، عدد، و پارامتر دوم، واحد زمانی را می‌پذیرد.

```
DateTimeOffset startTime = DateBuilder.FutureDate(2, IntervalUnit.Second);
```

در اینجا، زمان آغاز وظیفه را 2 ثانیه پس از آغاز برنامه تعریف کرده ایم. واحدهای زمانی دیگر شامل میلی ثانیه، دقیقه، ساعت، روز، ماه، هفته و سال هستند. کلاس DateBuilder، متدهای مختلفی برای تعیین زمان را در اختیار قرار می‌دهد. تعیین زمان آغاز به روش دیگر را به صورت کامنت شده در خط 13 مشاهده می‌کنید.

وظیفه‌ی ایجاد شده در خط 16 تا 18 معرفی شده است.

```
IJobDetail job = JobBuilder.Create<HelloJob>()
    .WithIdentity("job1")
    .Build();
```

پشتیبانی Quartz.NET از سینتکس fluent، کدنویسی را ساده و لذت بخش می‌کند. با استفاده از متد Create کلاس JobBuilder، وظیفه را معرفی می‌کنیم. متد Create، یک متد Generic است که نام کلاسی که اینترفیس IJob را پیاده سازی کرده است می‌پذیرد. یک نام را با استفاده از متد WithIdentity به وظیفه نسبت می‌دهیم (البته این کار، اختیاری است) و در انتها، متد Build را فراخوانی می‌کنیم. خروجی متد Build، از نوع IJobDetail است. و حالا نوبت به تنظیمات زمان رسیده است. در Quartz.NET، این مرحله، "ایجاد trigger" نام دارد. خطوط 20 تا 24 به این کار اختصاص دارند.

```
ITrigger trigger = TriggerBuilder.Create()
    .WithIdentity("trigger1")
    .StartAt(startTime)
    .WithSimpleSchedule(x =>
        x.WithIntervalInSeconds(10).WithRepeatCount(2))
    .Build();
```

ابتدا متد Create کلاس TriggerBuilder را فراخوانی می‌کنیم، سپس با استفاده از متد WithIdentity، یک نام به trigger اختصاص می‌دهیم (البته این کار، اختیاری است). با متد StartAt، زمان شروع وظیفه را که در ابتدا با استفاده از کلاس DateBuilder ایجاد کردیم تعیین می‌کنیم. مهمترین قسمت، تعیین دفعات و فواصل زمانی اجرای وظیفه است. همان طور که احتمالاً حدس زده اید، Quartz.NET مجموعه ای غنی از روش‌های مختلف برای تعیین بازه‌ی زمانی اجرا را در اختیار قرار می‌دهد. آسان‌ترین راه، استفاده از متد WithSimpleSchedule است. با استفاده از یک عبارت Lambda که ورودی آن از نوع کلاس SimpleScheduleBuilder است، دفعات و فواصل زمانی اجرا را تعیین می‌کنیم. متد WithIntervalInSeconds، برای تعیین فواصل زمانی در بازه‌ی ثانیه استفاده می‌شود. متد WithRepeatCount نیز برای تعیین دفعات اجرا است. وظیفه‌ی ما، 3 مرتبه و در فواصل زمانی 10 ثانیه اجرا می‌شود. مطمئن باشید اشتباه نکردم! بله، سه مرتبه. تعداد دفعات اجرا برابر است با عددی که برای متد WithRepeatCount تعیین می‌کنید، به علاوه‌ی یک. منطقی است، چون مرتبه‌ی اول اجرا زمانی است که با استفاده از متد StartAt تعیین کرده اید. در پایان، متد Build را فراخوانی می‌کنیم. خروجی متد Build، از نوع ITrigger است. آخرین کار (خطوط 26 تا 30)، ایجاد شی از اینترفیس IScheduler، فراخوانی متد ScheduleJob آن، و پاس دادن اشیای job و trigger که در قسمت قبل ایجاد شده اند به این متد است. در انتها، متد Start() را برای آغاز وظیفه فراخوانی می‌کنیم.

```
ISchedulerFactory sf = new StdSchedulerFactory();
IScheduler sc = sf.GetScheduler();
sc.ScheduleJob(job, trigger);
sc.Start();
```

حال شما یک وظیفه تعریف کرده اید که در هر جای برنامه به صورت زیر، قابل فراخوانی است.

```
ISchedule myTask = new HelloSchedule();
myTask.Run();
```

کتابخانه ای که با آن سر و کار داریم بسیار غنی است و امکانات بسیاری دارد. در قسمت بعد، با برخی امکانات دیگر این کتابخانه آشنا می‌شوید.

نظرات خوانندگان

نویسنده: محمد
تاریخ: ۱۳۹۱/۰۵/۲۴ ۱۳:۱۱

با سلام و تشکر از مطب مفید شما

من در مورد کارهای زمان بندی شده قبلا سرچ کردم و به این نتیجه رسیدم که دات نت تا قبل از ورژن 4، استاندارد برای انجام کارهای زمان بندی شده نداشت ولی در ورژن 4 فضای نام `system.threading.tasks.taskscheduler` را ارائه داده است. در صورت امکان تفاوت‌های میان این استاندارد و مطالب فوق را بیان کنید

با تشکر مجدد

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۲۴ ۱۵:۴۴

`System.Threading.Tasks.TaskScheduler`، یک کلاس `abstract` هست. باید برای اون پیاده سازی صورت بگیره و به خودیه خود، کار خاصی انجام نمیده. برای اجرای وظایف به طور همزمان یا ایجاد یک صف از وظایف استفاده میشه، یا میشه با استفاده از اون، `Thread` جاری رو ذخیره کرد تا بعداً بشه در هنگام پایان یافتن کار یک `Thread`، به کنترل‌های UI از طریق اون دسترسی داشت. در مبحث این مقاله جایی نداره.

نویسنده: محمد
تاریخ: ۱۳۹۱/۰۵/۲۴ ۱۶:۴۱

ممنون از شما

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱:۱۰

جناب مهندس راد سلام

روشی که اینجا شما فرمودین نمونش قبلا حدود 4 سال پیش من دیدم که آقای کیوان نیری توی سایتشون آموزش داده بودن (روشی شبیه همین روش با انعطاف پذیری بالا) اما مشکلی که توی این روش برای وب وجود داره اناجام کارهایی هست که باید حتما انجام بشه مثل واریز سود بانکی، در صورتی که سرور ری استارت بشه اطلاعات جاب‌های ما از بین میره که البته من خودم روش آقای نیری با دیتابیس ترکیب کردم و میشه گفت بدون اشکال چند سالی هست داره کار میکنه اگر این روش که فرمودین راهی داره برای ری استارت شدن وب ممنون می‌شم شرح بدین

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۲۵ ۹:۱۳

سلام برادر.

Quartz.NET رو دست کم نگیرید. بسیار بسیار قدرتمنده. این کتابخانه برای ردیابی وظایف از مفهومی با عنوان `JobStore` استفاده می‌کنه. به طور پیش فرض داده‌های مربوط به ردیابی وضعیت اجرای وظیفه‌ها در حافظه‌ی RAM قرار می‌گیرند که Quartz.NET اون رو به عنوان `RAMJobStore` میشناسه. اما برای سناریوهایی همانند آنچه که شما گفتید، پشتیبانی از پیش تعبیه شده برای ذخیره‌ی داده‌های ردیابی رو در پایگاه داده هم فراهم می‌کنه و این قابلیت رو با عنوان `AdoJobStore` میشناسه. این کتابخانه در حال حاضر از پایگاه‌های داده‌ی `SQLite`، `MySQL`، `Oracle`، `SqlServer` و `Firebird` پشتیبانی می‌کنه. فایل اسکریپت برای تولید جداول مورد نیاز در بسته‌ی داندودی اون هست.

در ضمن، Quartz.NET با خودش سرویسی به همراه دارد که میتونه به عنوان سرویس‌های سیستم عامل معرفی بشه تا هرگاه سیستم بالا اومد، اون سرویس به طور خودکار وظیفه‌ها رو از یک فایل با فرمت XML میخونه و اجرا می‌کنه.

در مورد 4 سال پیش که فرمودید، بله بنده هم پیاده سازی‌های مختلفی دیدم. حتی یکی از هموطنان، با استفاده از Cache، برای برای برنامه‌های ASP.NET، قابلیت زمانبندی (هر چند بسیار محدود) ارائه داده بود. استفاده از Timer هم روتین‌ترین چیزی هست که به ذهن همه میرسه اما در قسمت [پرسش و پاسخ](#) سایت Quartz.NET، دلایل مطلوب نبودن استفاده‌ی صرف از Timer در قسمت "Why not just use System.Timers.Timer?" گفته شده.

نویسنده: صابر فتح الهی
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۰:۳۶

ممنونم از پاسخ شما
فکر کنم از کد تایمر منظورتون پیاده‌سازی آقای سالار خلیلی هست دیگه؟
اگه اینجوری باشه خیلی عالیه مستنداتشو خوندم جالبه
ممنون از پست مفیدتون

نویسنده: بهمن خلفی
تاریخ: ۱۳۹۱/۰۵/۲۵ ۱۲:۴۳

با سلام و عرض ادب و سپاس فراوان
جناب راد از حوصله ای که به خرج دادید و مطلب درخواستی را با توضیحات بسیار کامل مطرح نمودید تشکر و سپاسگذارم
من قبلا از کد آقای نیری استفاده کردم خوب بود ولی کافی نیست یه سری اشکالات ریزی دارد که همیشه برای کارهای بزرگ و حساس استفاده کرد
ولی با خواندن مطلب شما و مراجعه به رفرنس مطلب مطمئن شدم که این امکان بسیار قدرتمند و قابل مقایسه با موارد مشابه خودش نیست
امیدوارم که همواره کامیاب باشید

نویسنده: علی معصومی
تاریخ: ۱۳۹۱/۰۵/۲۵ ۲۳:۵۷

سلام من یک وبسایتی دارم که در فواصل زمانی مشخص از وب سرویس پیامک‌های واردر رو دریافت میکنه و روی اونا پردازش انجام میده ادر حال حاضر این کارو دارم به وسیله یک سرویس و تایمر روی یک سرور اختصاصی انجام میدم که هزینه بالایی داره
برام به نظر شما این کارو میتونم با همین که شما آموزش دادید انجام بدم؟

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۵/۲۶ ۱۱:۱۴

بله، می‌تونید.

نویسنده: مهران
تاریخ: ۱۳۹۱/۰۷/۱۰ ۱۶:۵۳

سلام
یک کلاس با اسم Abzar وجود دارد میخوامم متدی که با اسم ADD() در این کلاس وجود دارد در هر پنج دقیقه یک بار اجرا شود.
توجه داشته باشید که در Page-Load نباید باشد.
و این برنامه برای یک وب سایت است

کاربر هیچ کاری با این کلاس و متد ندارد و هیچ وقت در صفحه ای باز نمیشود. ولی این متد در طول شبانه روز در هر 5 دقیقه یک بار اجرا میشود

آیا با Quartz.NET میتوان این کار را انجام داد ؟

نویسنده: بهروز راد
تاریخ: ۱۱:۲۱ ۱۳۹۱/۰۷/۱۱

شما باید از قوانین استفاده از این کتابخانه پیروی کنید. پیاده سازی اینترفیس های لازم برای استفاده از این کتابخانه ضروری هست...

کاری که قصد دارید انجام بدید کمترین کاریه که Quartz.NET می تونه انجام بده. در نقطه ی آغاز برنامه (روال Application_Start فایل Global.asax)، می تونید تعیین کنید که شروع اجرا در تاریخ و ساعت خاصی باشه و در بازه های زمانی مشخصی اجرا بشه. ضمناً، این کتابخانه از [فرمت cron](#) هم برای تعیین زمان پشتیبانی می کنه که اون رو خیلی قدرتمند می کنه.

نویسنده: مهران
تاریخ: ۲۱:۱ ۱۳۹۱/۰۷/۱۱

این کتابخانه کمی ناشناخته است و مطلب زیادی در موردش پیدا نکردم، آن چیزی هم که هست بیشتر انگلیسی است که بیشتر گیج کننده است.

اگر لطف کنید دقیق تر (با جزییات) بفرمایید چه کار باید بکنم ممنون میشوم .

یا مثالی با توجه به توضیحی که دادم بگذارید خیلی ممنون میشوم.

نویسنده: بهروز راد
تاریخ: ۹:۳۲ ۱۳۹۱/۰۷/۱۲

به نظر من، دقیق تر از این امکان توضیح وجود نداره.

نویسنده: جلال
تاریخ: ۲۰:۲۰ ۱۳۹۱/۰۸/۱۵

سلام.

ممنون بابت معرفی این کتابخانه ی مفید. جایی فرمودید که « در اینترفیس IJob در ASP.NET، به شی HttpContext دسترسی ندارید. » که درست نیست. شما در همه جای برنامه ASP.NET به کمک HttpContext.Current به HttpContext جاری دسترسی دارید. آیا استفاده از این روش مشکل خاصی داره؟

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۲ ۱۳۹۱/۰۸/۱۵

نه در همه جای یک برنامه ASP.NET. در یک سری از روال های global.asax.cs دسترسی به HttpContext وجود ندارد. روال هایی که دخالتی در پردازش درخواست وب جاری ندارند ([هرجایی که](#) Request context وجود نداشته باشد). از همین روال ها هم اتفاقاً برای اجرا و آغاز jobها در وب استفاده می شوند.

نویسنده: mahdi1391
تاریخ: ۱۶:۴۷ ۱۳۹۱/۰۹/۱۲

با سلام خدمت دوستان و اساتید عزیز. ممنون از این مقاله ارزشمند، واقعاً عالی و کاربردی است. من دقیقاً با مشکلی که در یکی از بندهای این کامنت (جناب بهروز راد) به آن اشاره کردند، مواجه هستم. مشکلی که من دارم در Shared Server است.

فرض کنید یک وظیفه در حال اجرا است، و در هاست های اشتراکی چندین بار در روز اتفاق می افتد که IIS ریستارت شود، Recycling انجام شود یا حتی سرور ریستارت شود؛ لطفأ در این مورد بیشتر توضیح بفرمائید که چطور آن وظیفه را دوباره به اجرا

دریابورم.

فعلاً مجبورم آن وظیفه را دستی فراخوانی کنم.

ممنون.

نویسنده: وحید نصیری

تاریخ: ۱۷:۲۳ ۱۳۹۱/۰۹/۱۲

یک فید برای سایت درست کنید. این فید را در فیدبرنر ثبت کنید. مابقی آن خودکار است. دیگر گوگل دست از سر شما برنخواهد داشت!

نویسنده: mahdi1391

تاریخ: ۲۰:۱۶ ۱۳۹۱/۰۹/۱۲

با تشکر از پاسخ شما.

شاید بهتر باشد مشکل را بگویم تا شاید سناریوی بهتری برای آن وجود داشته باشد.

در هاستهای اشتراکی (مخصوصاً هاستینگ‌های داخل کشور)، اگر مدت کوتاهی کسی به سایت شما سرزند، به صورت خودکار سایت از حافظه خارج میشود (که دلایل آن واضح است) و پس از درخواست نمایش صفحه، حداقل 30 ثانیه زمان لازم است تا دوباره چرخه اجرا انجام و صفحه اولیه نمایش داده شود. حال من دنبال راهی هستم که همیشه سایت در حال اجرا باشد. راه حل هایی [مثل این](#) را پیدا کردم که برای هاستهای اشتراکی کاربرد ندارد. ممنون.

نویسنده: محمد صاحب

تاریخ: ۲۰:۵۳ ۱۳۹۱/۰۹/۱۲

این [روش](#) رو تست کردی؟

نویسنده: وحید نصیری

تاریخ: ۲۰:۵۷ ۱۳۹۱/۰۹/۱۲

فیدبرنر ساعتی چندبار به سایت شما سر می‌زند. این سر زدن یعنی ارسال درخواست به سایت. در این حالت اگر سایت درون حافظه هم نباشد، مجدداً اجرا خواهد شد.

نویسنده: mahdi1391

تاریخ: ۲۱:۱۱ ۱۳۹۱/۰۹/۱۲

ممنون از جناب نصیری و جناب صاحب بابت پاسخ گویی.

تست میکنم.

با تشکر.

نویسنده: بهروز راد

تاریخ: ۱۰:۵ ۱۳۹۱/۰۹/۱۳

من از [Pingdom](#) استفاده می‌کنم. می‌تونید فواصل زمانی پینگ از سایت رو برای این سرویس روی 5 دقیقه تنظیم کنید.

نویسنده: محسن موسوی

تاریخ: ۱۴:۵۱ ۱۳۹۱/۰۹/۱۳

دوست عزیز من قبلاً از این روش استفاده کردم و جواب گرفتم: به این ترتیب که

در global

```
void Application_Start(object sender, EventArgs e)
```

```
{
    // Schedule Start
}

void Application_End(object sender, EventArgs e)
{
    // Code that runs on application shutdown
    var scheduler = new StdSchedulerFactory().GetScheduler();
    scheduler.Shutdown(true);
}

DoWebRequest ("SiteAddress");
}

public string DoWebRequest(string url)
{
    WebRequest request = WebRequest.Create(url);
    request.Credentials = CredentialCache.DefaultCredentials;
    HttpWebResponse response = (HttpWebResponse)(request.GetResponse());
    Stream dataStream = response.GetResponseStream();
    StreamReader reader = new StreamReader(dataStream);
    string responseFromServer = reader.ReadToEnd();
    reader.Close();
    dataStream.Close();
    response.Close();
    return responseFromServer;
}
```

نویسنده: مسعود زیانی
تاریخ: ۱۶:۲۰ ۱۳۹۱/۰۹/۱۴

با سلام و تشکر بابت مطلب عالیتون
من میخوام به تعداد بالا پیامک ارسال کنم و هر اگه هر کدوم ناموفق بود مثلا هر 3 ثانیه باز تا مثلا 4 بار تلاش کنه که باز پیامک رو ارسال کنه...اینکار رو با تایمر انجما میدم اما متاسفانه تو تعداد بالا سرورم استاپ میشه و فقط در تعدادهای کم مثلا زیر 300 تا درست کار میکنه
با این کلاس شما اینکار من کاملا قابل هندل هست؟

نویسنده: بهروز راد
تاریخ: ۷:۷ ۱۳۹۱/۰۹/۱۵

مشکل توقف Server شما ارتباطی با این بحث نداره و باید در روش و کدهایی که استفاده می کنید دلیل رو جستجو کنید.
اما برای ارسال زمانبندی شده ی SMS، بله می تونید از این کتابخانه استفاده کنید.

نویسنده: مسعود زیانی
تاریخ: ۲۳:۱۵ ۱۳۹۱/۰۹/۱۶

خیلی ممنون

نویسنده: محمد حسین فخرآوری
تاریخ: ۱۵:۲۳ ۱۳۹۱/۱۱/۲۳

با سلام
اقای راد چرا برنامه روی هاست بعد مدتی از کار میافته.

نویسنده: وحید نصیری
تاریخ: ۱۷:۲۵ ۱۳۹۱/۱۱/۲۳

لطفا [نظرات همین مطلب](#) رو از ابتدا مطالعه کنید؛ بحث شده قبلا.

نویسنده: محمد حسین فخرآوری
تاریخ: ۱۱:۵۵ ۱۳۹۱/۱۱/۲۴

اقای نصیری در سایت pingdom یا ثبتنامش پولی؟
و http://feedburner.google.com چطوری ثبتنام کنم؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۹ ۱۳۹۱/۱۱/۲۴

- فیدبرنر جزو خدمات گوگل است. همینقدر که یک اکانت گوگل یا جی میل دارید به [خدمات فیدبرنر](#) هم دسترسی خواهید داشت.
- برای سرویس‌های دیگر که به سایت ping ارسال می‌کنند، از [جستجوی گوگل](#) شروع کنید.

نویسنده: محمد حسین فخرآوری
تاریخ: ۱۳:۳۵ ۱۳۹۱/۱۱/۲۴

ممنون آقای نصیری .
یکی از دوستان روشی گفتن (publicstringDoWebRequest(stringurl) در Application_End
ایا این روش هم درسته و میتوان روش حساب کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳:۳۹ ۱۳۹۱/۱۱/۲۴

خیر. سرور می‌تونه ری استارت یا خاموش بشه. امکان خیلی از اتفاقات دیگر هم هست که نیاز به راه اندازی مجدد و رسیدن اولین درخواست رو داره. در کل روش ارسال ping به سرور از طریق یک برنامه خارجی مطمئن‌ترین است.

نویسنده: محمد حسین فخرآوری
تاریخ: ۱۳:۲۱ ۱۳۹۱/۱۱/۲۷

با سلام
دوستان اگر ممکن راهنمای ثبت سایت در این 2 سایت بالا که آقای نصیری گفتن بزارین.
هر کاری میکنم نمیشه

نویسنده: محسن
تاریخ: ۱:۳ ۱۳۹۱/۱۱/۲۸

از سایت [پالس‌می](#) هم می‌تون استفاده کنی.

نویسنده: علی
تاریخ: ۱۵:۲۷ ۱۳۹۲/۰۴/۳۱

بعد از اجرا این پیغام رو می‌ده:

Could not load file or assembly 'Common.Logging' or one of its dependencies. This assembly is built by a runtime newer than the currently loaded runtime and cannot be loaded

نویسنده: وحید نصیری
تاریخ: ۱۶:۵۷ ۱۳۹۲/۰۴/۳۱

پیغام خطایی که دریافت کردید به این معنا است که اسمبلی دریافتی با یک شماره بالاتر دات نت کامپایل شده و در شماره نگارش مدنظر شما که پایین‌تر هست قابل استفاده نیست.
بهتر است سورس کد کتابخانه را دریافت و کامپایل یا اسمبلی‌های قدیمی‌تر این کتابخانه را دریافت کنید.

بنده یک مشکل عجیب با quartz پیدا کردم
من یک schedule برای ایمیل نوشتم که بتونه ایمیل انبوه با فاصله زمانی ارسال کند

منتها من برای اینکه ارسال ایمیل شود باید لیست ایمیل ها را در هر بار به job بفرستم و با هر بار فرستادن , آن ایمیل از لیست ایمیل ها پاک شود . برای اینکه بتوانم لیست ایمیل ها را با هر بار اجرای job حفظ کنم که متوجه شوم چه ایمیل هایی مانده است از اتربیوت PersistJobDataAfterExecution و DisallowConcurrentExecution بالای سر job استفاده کردم .
در job گفتم اگر تعداد لیست ایمیل ها به صفر رسید schedule متوقف شود
در لوکال مشکلی ندارد ولی در عملی متوجه شدم گویا مقدار لیست ایمیل ها حفظ نمی شود و مجدد ایمیل زده می شود. لطفا کمک کنید

```
[PersistJobDataAfterExecution]
[DisallowConcurrentExecution]
public class SendGroupEmailJob : IJobBase
{
    private List<MailAddress> lstMails;

    public void Execute(IJobExecutionContext context)
    {
        int result = 0;
        if (context.JobDetail.JobDataMap["UserEmailList"] != null)
        {
            lstMails = context.JobDetail.JobDataMap["UserEmailList"] as List<MailAddress>;

            if (lstMails.Count == 0)
            {
                context.Scheduler.UnscheduleJob(new TriggerKey(context.Trigger.Key.Name));
            }
            else
            {
                JobDataMap map = context.JobDetail.JobDataMap;
                result = EmailHandler.Send(lstMails[0],
                    map.GetString("Subject"),
                    map.GetString("Body").Replace("[FullName]",
lstMails[0].DisplayName).Replace("[Email]", lstMails[0].Address),
                    context.JobDetail.JobDataMap["Attachment"] as List<string>,
                    MailPriority.High,
                    true,
                    Encoding.UTF8,
                    DeliveryNotificationOptions.None,
                    map.GetString("SenderEmail"),
                    map.GetString("SenderName"),
                    map.GetString("BccEmail"),
                    map.GetString("Prefix"),
                    map.GetBoolean("IsSSL"),
                    map.GetBoolean("IsCredential"),
                    map.GetString("Server"),
                    map.GetInt("Port"),
                    map.GetInt("TimeOut"),
                    map.GetString("PassWord"));
                lstMails.RemoveAt(0);
            }
        }
    }
}
```

1stMails رو [استاتیک](#) تعریف کن. به علاوه هنگام بررسی count آن یک lock هم باید قرار بدی. ضمناً بررسی count آن را ببر قبل از انتساب به آن. الان داری یک لیست کامل رو به یک لیست به ظاهر موجود انتساب می‌دی بعد مقدار تعداد آیتم‌هاش رو حساب می‌کنی. خوب، این مقدار همیشه مساوی است با تعداد لیست جدید انتساب داده شده. همچنین برنامه‌های وب ممکن است [به دلایل زیادی](#) ری‌استارت شوند؛ یعنی از دست رفتن تمام متغیرهای درون حافظه. بنابراین بهتر است از دیتابیس استفاده کنید.

نویسنده: محسن
تاریخ: ۱۳۹۲/۰۹/۰۳ ۱۲:۰۰

لطفاً یک نمونه سمپل بذارید

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۰۳ ۱۲:۲۲

یک سری به سورسش بزنید. [پیر است از مثال](#) .

نویسنده: عصارى
تاریخ: ۱۳۹۲/۱۰/۲۶ ۱۴:۴۳

با سلام؛ من برای ارسال پیامک‌های زمانبندی شده، از همین روش استفاده کردم، اما یک مشکل دارم. اونم اینه که در صورتی که چند job به طور همزمان اجرا بشن تداخل پیدا میکنن. برای توضیح بیشتر، زمانی که چند job رو با عبارت cron یکسان ثبت میکنم -> مثلاً:

? * * 10 33 0

پس از اجرا و ارسال پیام که باید تراکنش مالی اون ثبت بشه (لازم به توضیح است که من عملیات ارسال پیام و ثبت تراکنشات مالی رو در متد execute نوشتم)، ثبت نمیشه و اجرای بقیه jobها ادامه پیدا میکنه. ممنون میشم بنده رو راهنمایی کنید.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۷ ۱:۳۱

به صورت پیش فرض جاب‌ها به صورت موازی پردازش می‌شوند. اگر می‌خواهید سریالی شوند از ویژگی DisallowConcurrentExecution استفاده کنید (روی نام کلاس قرارش بدید).

نویسنده: عصارى
تاریخ: ۱۳۹۲/۱۰/۲۸ ۱۰:۰۱

من از (DisallowConcurrentExecution) استفاده کردم اما باز jobها به صورت موازی اجرا میشن!؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۸ ۱۰:۱۹

طراحی lock و مباحث همزمانی (lock, mutex, semaphore) و Threading را باید خودتان پیاده سازی کنید. [DisallowConcurrentExecution](#) یعنی یک وهله از یک کلاس خاص در زمان اجرا ایجاد شده و چندین وهله همزمان از آن ایجاد نخواهد شد. [روش دیگر اینکار](#) استفاده از TriggerListener, JobListener or SchedulerListener است. یکی که تمام شد، بعدی شروع شود.

نویسنده: بهاره فیضی
تاریخ: ۱۳۹۳/۰۳/۲۹ ۱۱:۴۰

سلام؛ من سعی کردم این کدها رو تست کنم بنابراین قدم به قدم پیش رفتم. بعد وظیفه رو به شکل زیر فراخوانی کردم

```
protected void Application_Start(object sender, EventArgs e)
{
    Interface1 myTask = new HelloSchedule();
    myTask.Run();
}
```

دو بار web application رو اجرا کردم و نتیجه به شکل زیر بود

```
Message from HelloJob 6/19/2014 10:40:15 AM
Message from HelloJob 6/19/2014 10:40:25 AM
Message from HelloJob 6/19/2014 10:40:35 AM
Message from HelloJob 6/19/2014 10:40:45 AM
Message from HelloJob 6/19/2014 10:40:55 AM
Message from HelloJob 6/19/2014 11:24:36 AM
Message from HelloJob 6/19/2014 11:24:45 AM
Message from HelloJob 6/19/2014 11:24:55 AM
```

یعنی دفعه اول 5 بار وظیفه اجرا شده دفعه دوم 3 بار. ولی هدف من از اعمال زمانبندی اینه که بی نهایت بار وظیفه اجرا بشه، به طور مثال در یک سایت آگهی شبیه ایستگاه، مثلاً دو روز قبل از انقضای آگهی به فرد اطلاع داده بشه که آگهی شما داره منقضی میشه. البته من در لوکال تست کردم. من بهتر از

Application_Start

جایی برای فراخوانی پیدا نکردم. لطفاً راهنمایی کنید.

نویسنده: محبوبه عصارى
تاریخ: ۱۳۹۳/۰۴/۱۱ ۱۰:۴۸

با سلام
ممنون از راهنماییتون، ممکن هست برای استفاده از joblistener مثالی بذارید؟ برای ارسال پیامک به صورت ترتیبی چطور باید این لیست رو به joblistener اضافه نمود و چطور باید بعد از هر ارسال ، ارسال بعدی را فراخوانی نمود؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۱۱:۳۳

برای ارسال پیامک به صورت ترتیبی از حلقه استفاده کنید. زمانیکه job اجرا شد، ابتدا لیست اشخاصی را که باید پیامک دریافت کنند از دیتابیس واکنش کرده و سپس در طی یک حلقه، به آنها پیام ارسال کنید.

نویسنده: محبوبه عصارى
تاریخ: ۱۳۹۳/۰۴/۱۱ ۱۲:۰۱

دقیقا این کاری بود که من قبلا انجام داده بودم ولی مشکلی که پیش می آمد این بود که مثلا اگر در یک ارسال بایستی 200 پیامک ارسال گردد قبل از اینکه ارسال این 200 پیامک به اتمام برسد زمان اجرا به پایان رسیده و تابع execute مجددا فراخوانی می شود و از انجایی که هنوز وضعیت این رکورد در دیتابیس به ارسال شده تغییر پیدا نکرده مجددا این 200 پیامک ارسال می گردد. این مشکل حتی زمانی که از حلقه for هم استفاده نمی شود وجود دارد و در تعداد ارسال بالا به مشکل می خورد . در زیر کدی که برای ارسال استفاده نموده ام را قرار دادم. با تشکر از شما

```
namespace SchedulerDemo.Jobs
{
    using System;
```

```

using System.Linq;
using System.IO;
using Quartz;
using System.Collections.Generic;
using System.Configuration;

[PersistJobDataAfterExecution]
[DisallowConcurrentExecution]
public class SendJob : IJob
{
    public void Execute(IJobExecutionContext context)
    {
        using (var db = new DALModel.DALEntities())
        {
            byte status = (byte)AllEnums.Sms.Status.InProgress;
            var item = db.SentBoxes.Where(p => p.Status == status && p.IsDeleted==false &&
p.UserInfo.IsDeleted==false && p.HasTime == true && p.SendInTime == false && p.SendDateX <=
DateTime.Now).OrderBy(p=>p.Id).FirstOrDefault();
            Cls_SMS.ClsSend sms_Batch = new Cls_SMS.ClsSend();

            if (item != null)
            {
                decimal smsCount = 0;
                if (item.UserInfo.CalculateType ==
Convert.ToByte(AllEnums.FinancialTransaction.CalculationUnit.Message))
                {
                    smsCount = Convert.ToDecimal(Function.GetSmsCount(item.Price, item.UserId));
                }
                else
                {
                    smsCount = Convert.ToDecimal(item.CorrectCount);
                }
                decimal adminCredit = Function.GetAdminCreditLink1000();
                if (adminCredit != -1 && adminCredit >= smsCount)
                {
                    if ((item.UserInfo.Credit - (item.UserInfo.LowCredit)) >= item.Price)
                    {
                        item.SendInTime = true;
                        db.SaveChanges();
                        string numberList = item.NumberList;
                        int position = item.NumberList.LastIndexOf(',');
                        numberList = item.NumberList.Substring(0, position);
                        List<string> receivers_List = new List<string>();
                        receivers_List = (numberList).Split(',').ToList();
                        string[] ret2 = new string[2];
                        string[] DestAdd = new string[receivers_List.Count];
                        DestAdd = receivers_List.ToArray();
                        ret2 = sms_Batch.SendSMS_Batch(item.Message, DestAdd,
item.UserInfo.SenderNumber.AllNumber.Number, ConfigurationManager.AppSettings["SmsUserNameLink1000"],
ConfigurationManager.AppSettings["SmsPasswordLink1000"],
ConfigurationManager.AppSettings["SmsIPAddressLink1000"],
ConfigurationManager.AppSettings["SmsCompanyLink1000"], false, item.Id);
                        var sentBoxUpdate = db.SentBoxes.FirstOrDefault(p => p.Id == item.Id);
                        sentBoxUpdate.Status = Convert.ToByte(AllEnums.Sms.Status.Send);
                        sentBoxUpdate.FinancialTransactionId = db.FinancialTransactions.Where(p =>
p.UserId == item.UserId).Max(p => p.Id);

                        if (ret2 != null)
                        {
                            sentBoxUpdate.RetValue0 = ret2[0];
                            sentBoxUpdate.RetValue1 = ret2[1];
                        }
                        db.SaveChanges();
                    }
                    else
                    {
                        byte statusFailedForAccount = (byte)AllEnums.Sms.Status.FailedForAccount;
                        item.SendInTime = true;
                        item.Status = statusFailedForAccount;
                        item.FailedCount = item.CorrectCount;
                        item.FailedList = item.NumberList;
                        db.SaveChanges();
                    }
                }
            }
            else
            {
                byte statusFaildForError = (byte)AllEnums.Sms.Status.FaildForError;

```

```

        item.SendInTime = true;
        item.Status = statusFaieldForError;
        item.FailedCount = item.CorrectCount;
        item.FailedList = item.NumberList;
        db.SaveChanges();
    }
}

}

}

}

namespace SchedulerDemo.Interfaces
{
    public interface IScheduleSend
    {
        void RunSendSms();
    }
}

namespace SchedulerDemo.Jobs
{
    using System;
    using Quartz;
    using Quartz.Impl;
    using SchedulerDemo.Interfaces;
    using SchedulerDemo.Jobs;

    public class SendSchedule : IScheduleSend
    {
        public void RunSendSms()
        {
            DateTimeOffset startTime = DateBuilder.FutureDate(2, IntervalUnit.Second);

            IJobDetail job = JobBuilder.Create<SendJob>()
                .WithIdentity("jobSendSmsInTime")
                .Build();

            ITrigger trigger = TriggerBuilder.Create()
                .WithIdentity("triggerSendSmsInTime")
                .StartAt(startTime)
                .WithSimpleSchedule(x =>
x.WithIntervalInMinutes(5).RepeatForever())
                .Build();

            ISchedulerFactory sf = new StdSchedulerFactory();
            IScheduler sc = sf.GetScheduler();
            sc.ScheduleJob(job, trigger);

            sc.Start();
        }
    }
}

```

نویسنده:

وحید نصیری

تاریخ:

۱۲:۱۴ ۱۳۹۳/۰۴/۱۱

از [DNT Scheduler](#) استفاده کنید. در اینجا اگر یک job هنوز در حال اجرا باشد، وهله‌ی جدیدی از آن آغاز نخواهد شد:

```
var taskToRun = _tasks.Where(x => !x.IsRunning && x.RunAt(now)).OrderBy(x => x.Order).ToList();
```

در قسمت IsRunning این بررسی به صورت خودکار انجام می‌شود.

نویسنده:

بهاره فیضی

تاریخ:

۱۱:۳ ۱۳۹۳/۰۵/۰۲

سلام.

من به وظیفه نوشتم که می‌باد هر 1 دقیقه 1 بار به متن رو در Table درج می‌کند.

```

using Quartz;
using System.Data;
using System.Data.OleDb;
using System.Configuration;

namespace WebApplication1
{
    public class TestQuartzClass:IJob
    {
        public void Execute(IJobExecutionContext context)
        {
            //Page myPage = (Page)HttpContext.Current.Handler;
            //TextBox MyTextBox=(TextBox)myPage.FindControl("txt");

            string sql = "Insert into tbl_Test (Content) values (@Content)";
            ExecuteNoneQuery(System.Data.CommandType.Text, sql, new OleDbParameter[]{
                //new OleDbParameter("@Content", MyTextBox.Text)
                new OleDbParameter("@Content","Hello world!")
            });
        }

        public int ExecuteNoneQuery(CommandType commandType, string commandText, params
OleDbParameter[] commandParameters)
        {
            using (OleDbConnection con = new
OleDbConnection(ConfigurationManager.ConnectionStrings["ConStr"].ConnectionString))
            {
                OleDbCommand cmd = new OleDbCommand();
                cmd.Connection = con;
                cmd.CommandType = commandType;
                cmd.CommandText = commandText;
                cmd.Parameters.AddRange(commandParameters);
                con.Open();
                int retVal = cmd.ExecuteNonQuery();
                con.Close();
                return retVal;
            }
        }
    }
}

```

و این هم کد صفحه ای که با کلیک دکمه وظیفه شروع به کار می‌کند

```

using System;
using Quartz;
using Quartz.Impl;

namespace WebApplication1
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        public static void ConfigureQuartzJobs()
        {
            // construct a scheduler factory
            ISchedulerFactory schedFact = new StdSchedulerFactory();

            // get a scheduler
            IScheduler sched = schedFact.GetScheduler();
            sched.Start();
            IJobDetail job = JobBuilder.Create<TestQuartzClass>()

```

```
        .WithIdentity("SendJob")
        .Build();
var trigger = TriggerBuilder.Create()
    .WithIdentity("SendTrigger")
    .WithSimpleSchedule(x => x.WithIntervalInMinutes(1).RepeatForever())
    //.StartAt(startTime)
    .StartNow()
    .Build();

    sched.ScheduleJob(job, trigger);
}

protected void btn_Click(object sender, EventArgs e)
{
    ConfigureQuartzJobs();
}
}
```

همونطور که می بینید متن رو به شکل زیر پاس دادم.

```
new OleDbParameter("@Content", "Hello world!")
```

ولی من می خوام این متن رو از TextBox بگیرم.

و برای اینکه در کلاس بتونم به کنترل های صفحه دسترسی داشته باشم کدهای زیر رو به متد Exceute اضافه کردم

```
Page myPage = (Page)HttpContext.Current.Handler;
TextBox MyTextBox=(TextBox)myPage.FindControl("txt");
```

ولی به محض اینکه این کدها رو اضافه می کنم دیگه برنامه کار نمی کنه.

متن داخل تکست باکس رو هم قصد داشتم به شکل زیر پاس بدم.

```
new OleDbParameter("@Content", MyTextBox.Text)
```

لطفاً راهنمایی کنید.

من یک نمونه هم به منظور تست آماده کردم که از لینک زیر می تونید دانلود کنید:

http://www.4shared.com/rar/1Fu_jp00ba/WebApplication1.html

نویسنده: محسن خان

تاریخ: ۱۱:۱۴ ۱۳۹۳/۰۵/۰۲

ترد اجرایی یک وظیفه ی پس زمینه با ترد اجرایی یک درخواست وب یکی نیست. بنابراین نمی تونید به اشیاء رابط کاربری در تردی دیگر دسترسی داشته باشید. در برنامه های دسکتاپ اگر این کار را انجام بدید، برنامه آنی کرش می کنه. در وب هم فرقی نمی کنه؛ اصول یکی هست.

کاری که می خواهید انجام بدید کلاً نباید از این طریق انجام بشه. اگر هدفتون مثلاً پیاده سازی **auto-save** هست که در بعضی از سایت ها دیدید که هر از چند ثانیه یکبار متن رو ذخیره می کنند، این رو با یک تایمر جاوا اسکریپتی سمت کاربر و Ajax انجام می دن و نه با یک وظیفه ی پس زمینه.

نویسنده: بهاره فیضی

تاریخ: ۱۲:۳۵ ۱۳۹۳/۰۵/۰۲

ممنون که پاسخ دادید.

نه من این رو تستی انجام دادم.

هدف اصلی من ارسال ایمیل انبوه بود به این صورت که تعداد زیادی ایمیل بر اساس صنف‌های کاری در یک جدول ذخیره شدن. مثلاً صنف ساختمانی، ...

حالا من می‌خواهم یک صفحه داشته باشم

ادمین بیاد از دراپ دان گروه رو انتخاب کنه و بر حسب صنف حالا مثلاً 100 تا ایمیل بهش نشون داده میشه.

بعد به فرض یه متن رو در یک تکست باکس وارد کنه و این متن به این افراد ایمیل بشه.

در این لینک توضیحات کاملتری رو داده بودم ولی هنوز جوابی نگرفتم.

<http://forums.asp.net/t/1998923.aspx?send+mass+email+using+quartz+net>

از ویندوز سرویس هم نمی‌تونم استفاده کنم چون هاستها اشتراکی هستند و ادمین هاست اجازه نصب نمی‌ده.

نویسنده:

محسن خان

تاریخ:

۱۳۹۳/۰۵/۰۲ ۱۲:۴۱

یک جدول طراحی کنید به نام mass mail که در آن یک Content به علاوه فیلد IsDone وجود داره. مدیر سیستم متن خودش رو به صورت معمول در این جدول ثبت کنه. در وظیفه‌ی پس زمینه، رکوردهایی را که IsDone آن‌ها false است یکی یکی یافته و پردازش کنید. بعد از اتمام کار، IsDone هر رکورد را true کنید.

مزیت این روش اینه که دو ترد کاملاً مجزای رابط کاربری و ترد وظیفه‌ی پس زمینه، هیچ تداخل و تماسی با هم نخواهند داشت تا سبب کرش برنامه شوند.

نویسنده:

بهاره فیضی

تاریخ:

۱۳۹۳/۰۵/۰۲ ۱۲:۵۶

واقعاً از تون ممنونم. چه روش خوب و ساده ای گفتید. اصلاً به ذهن خودم نرسیده بود.

نویسنده:

بهاره فیضی

تاریخ:

۱۳۹۳/۰۵/۱۲ ۲۲:۱۷

سلام.

من کدهای زیر رو در رویداد کلیک یک دکمه نوشتم که کاربر با کلیک دکمه اقدام به ارسال ایمیل به صورت گروهی می‌کنه.

حالا یکی از کاربرها با این ارور مواجه شده

.Unable to store Job: 'DEFAULT.SendJob', because one already exists with this identification

Description: An unhandled exception occurred during the execution of the current web request. Please review the .stack trace for more information about the error and where it originated in the code

Exception Details: Quartz.ObjectAlreadyExistsException: Unable to store Job: 'DEFAULT.SendJob', because one .already exists with this identification

:Source Error

.Build: 374: Line();

375: Line

376: sched.ScheduleJob(job, trigger);

377: Line

که البته من متوجه شدم دلیل این به خاطر اینه که هنوز جاب قبلی پایان نیافته کاربر دوباره روی دکمه کلیک می‌کنه و چون هنوز جاب قبلی اتمام نیافته با این ارور مواجه میشیم. بنابراین من یک دکمه گذاشتم و کدهای زیر رو نوشتم در رویداد کلیک این دکمه

```
var scheduler = new StdSchedulerFactory().GetScheduler();
scheduler.DeleteJob(new JobKey("SendJob"));
```

که با کلیک این دکمه جاب قبلی متوقف میشه و دیگه اون ارور رو نمی‌گیریم. ولی مشکل اینجاست که چطور میشه بدون اینکه اصلاً با اون ارور کاربر مواجه بشه پیغام بدیم که شما در حال حاضر مجاز به ارسال نیستید و مثلاً 10 دقیقه بعد دوباره اقدام به ارسال فرمایید.

یا اینکه چطور میشه یک جاب که تموم شد به کاربر پیغام بدیم که جاب پایان یافته، در این صورت دیگه کاربر در زمان نامناسب اقدام به ایجاد مجدد جاب نخواهد کرد.

ممنون.

نویسنده: بهاره فیضی
تاریخ: ۱۳۹۳/۰۵/۱۳ ۱۳:۲۱

فعلاً با کد زیر مشکل موقتاً حل شد، یعنی کاربر باید با کلیک روی دکمه مطمئن بشه که حالا می‌تونه اقدام به ارسال بعدی کنه یا نمی‌تونه. جالبه متوجه شدم اون کدهای متوقف کردن جاب هم در جا عمل نمی‌کنن، یعنی معلوم نیست کی متوقف بشه. پس اینجا به درد کار ما نمی‌خوره. اگه راه بهتری به نظرتون رسید ممنون می‌شم در میون بزارید.

```
protected void Button6_Click(object sender, EventArgs e)
{
    var scheduler = new StdSchedulerFactory().GetScheduler();
    var jobs = scheduler.GetCurrentlyExecutingJobs();
    //foreach (var j in jobs)
    //{
        Console.WriteLine("Progress of {0} is {1}",
            j.JobDetail.Key,
            j.JobDetail.JobDataMap["progress"]);
        Label620.Text += j.JobDetail.Key.ToString();
    //}
    if (jobs.Count == 0)
    {
        Label620.Text = "!آماده برای ارسال ایمیل";
    }
    else
    {
        Label620.Text = "!لطفاً بعد از چند دقیقه مجدداً تلاش نمایید";
    }
}
```

در این قسمت، نحوه‌ی استفاده از قابلیت‌های کتابخانه‌ی Quartz.NET را در قالب پرسش و پاسخ ادامه می‌دهیم.

ابتدا یک توضیح کلی:

برای مدیریت وظیفه‌ها در Quartz.NET، در هر جای پروژه می‌توانید به صورت ذیل به مدیر وظیفه‌ها دسترسی داشته باشید.

```
var scheduler = new StdSchedulerFactory().GetScheduler();
```

از حالا به بعد، هر جا که در کدها کلمه‌ی scheduler را دیدید، ایجاد آن از طریق خط قبل بوده است.

سعی کنید همیشه هنگام ایجاد اشیا از نوع IJobDetail و ITrigger، از متد WithIdentity (همان طور که در [قسمت قبل](#) مشاهده کردید) برای نامگذاری وظایف و triggerها استفاده کنید تا بتوانید بعداً با استفاده از نامشان به آنها ارجاع پیدا کرده و مدیریتشان کنید

1) سوال: چگونه می‌توان در یک زمان دلخواه (مثلاً در زمان کلیک بر روی یک دکمه)، اجرای یک وظیفه را متوقف کرد؟
جواب: برای توقف تمامی وظایف می‌توان از متد Shutdown() شی scheduler استفاده کرد:

```
scheduler.Shutdown(true);
```

در صورتی که مقدار true را به متد Shutdown پاس دهید، تا زمانی که وظایفی که در وسط اجرای خود هستند کارشان به پایان نرسد، کنترل اجرای برنامه به خط بعد نمی‌رود، اما در صورتی که این متد را بدون پارامتر فراخوانی کنید یا مقدار false را به آن پاس دهید، کنترل اجرای برنامه به دستور بعد می‌رود و وظایف در پشت صحنه به کار خود ادامه می‌دهند. دقت داشته باشید که منظور از ادامه‌ی کار یک وظیفه، ادامه‌ی کار آن در وضعیت جاری خود است. به بیان واضح تر، اگر مرتبه‌ی اجرای یک وظیفه 20 مرتبه بود و در مرتبه‌ی دوم اجرای آن، متد Shutdown() به هر صورتی فراخوانی شد، مرتبه‌های دیگر به هیچ وجه اجرا نمی‌شوند. اگر چند وظیفه به طور همزمان در حال اجرا باشند و قصد داشته باشید تا یکی از آنها را متوقف کنید، یکی از دو حالت زیر وجود دارد:

1) یک وظیفه به چند trigger نسبت داده شده است.

2) هر وظیفه فقط یک trigger دارد.

در صورتی که قصد دارید وظیفه از تمامی triggerها گرفته شود (معمولاً هم همین رفتار مد نظر است)، از متد DeleteJob استفاده کنید؛ و اگر قصد دارید تا اجرای وظیفه توسط یک trigger مشخص لغو شود و triggerهای دیگر مختص آن وظیفه به کار خود ادامه دهند، از متد UnscheduleJob استفاده کنید. اگر از متد DeleteJob استفاده می‌کنید، نام وظیفه را با ایجاد نمونه ای از کلاس JobKey برای آن مشخص کنید و در صورتی که از متد UnscheduleJob استفاده می‌کنید، نام trigger را با ایجاد نمونه ای از کلاس TriggerKey تعیین کنید.

```
scheduler.DeleteJob(new JobKey("job1"));
// or
scheduler.UnscheduleJob(new TriggerKey("trigger1"));
```

2) سوال: چگونه می‌توان اجرای وظیفه‌ها را به حالت تعلیق در آورد؟

جواب: برای به تعلیق در آوردن اجرای تمامی وظایف، از متد Standby() استفاده کنید:

```
scheduler.Standby();
```

برای ادامه‌ی کار، متد Start() را مجدداً فراخوانی کنید.

در صورتی که قصد دارید اجرای وظیفه ای خاص را به حالت تعلیق در آورید، از متد `PauseJob()` استفاده کنید. نام وظیفه را با ایجاد نمونه ای از کلاس `JobKey` برای آن مشخص کنید:

```
scheduler.PauseJob(new JobKey("job1"));
```

برای ادامه ی وظیفه، از متد `ResumeJob()` استفاده کنید. نام وظیفه را با ایجاد نمونه ای از کلاس `JobKey` برای آن مشخص کنید:

```
scheduler.ResumeJob(new JobKey("job1"));
```

برای تعلیق اجرای تمامی وظایف، متد `PauseAll()`، و برای ادامه ی کار تمامی وظایف، متد `ResumeAll()` را فراخوانی کنید.

(3 سوال: چگونه می توان یک وظیفه ی در حال اجرا را آپدیت کرد و تغییر مشخصات داد؟
جواب: با استفاده از متد `AddJob` و تنظیم پارامتر دوم آن به مقدار `true`:

```
IJobDetail job = JobBuilder.Create<NewJob>()  
    .WithIdentity("job1")  
    .Build();  
  
scheduler.AddJob(job, true);
```

اگر قبلاً کلاسی با عنوان `OldJob` برای وظیفه ای با نام `job1` تعریف شده است، با استفاده از قطعه کد بالا می توان کلاس `NewJob` را به جای آن معرفی کرد. البته به شرطی که نام وظیفه ی جدید با نام وظیفه ی قدیم، یکسان باشد. پارامتر دوم متد `AddJob` مشخص می کند که آیا در صورتی که نام وظیفه ای که قرار است در فرایند زمان بندی قرار بگیرد با نام یکی از وظایف موجود یکسان باشد، وظیفه ی جدید، جایگزین وظیفه ی قدیم شود یا خیر.

(4 سوال: چگونه می توان یک `trigger` در حال اجرا را آپدیت کرد و تغییر مشخصات داد؟
جواب: یک `trigger` جدید ایجاد و با استفاده از متد `RescheduleJob()`، جایگزین `trigger` قدیمی کنید:

```
ITrigger trigger = TriggerBuilder.Create()  
    .WithIdentity("newTrigger")  
    .StartNow()  
    .ForJob("job1")  
    .WithSimpleSchedule(x =>  
x.WithIntervalInSeconds(5).WithRepeatCount(20))  
    .Build();  
  
scheduler.RescheduleJob(new TriggerKey("oldTrigger"), trigger);
```

نام `trigger` جدید می تواند با نام `trigger` قدیم یکسان باشد. در تکه کد قبل، `trigger` یی با نام `newTrigger` ایجاد و زمان اجرای آن به حال تنظیم شده است. با استفاده از متد `ForJob()` و تعیین نام وظیفه، `trigger` جدید را به وظیفه ای با نام `job1` نسبت داده ایم. بازه ی زمانی اجرا، هر 5 ثانیه و 21 مرتبه خواهد بود. در متد `RescheduleJob()` و در پارامتر اول آن، نام `trigger` قدیمی را با ایجاد شی ای از کلاس `TriggerKey` مشخص کرده ایم و به پارامتر دوم، شی ایجاد شده برای `trigger` جدید را پاس داده ایم.

(5 سوال: چگونه می توان تعداد دفعات اجرای یک وظیفه را بی نهایت تعیین کرد؟
پاسخ: با استفاده از متد `RepeatForever()` در هنگام ایجاد `trigger`:

```
ITrigger trigger = TriggerBuilder.Create()  
    .WithIdentity("trigger1")  
    .StartAt(startTime)  
    .ForJob("job1")  
    .WithSimpleSchedule(x => x.WithIntervalInSeconds(5).RepeatForever())  
    .Build();
```

(6 سوال: چگونه می توان تعداد دفعات اجرای تمامی وظایف را به دست آورد؟

پاسخ: با استفاده از متد `GetMetaData()`:

```
SchedulerMetaData metaData = scheduler.GetMetaData();  
int numberOfJobsExecuted = metaData.NumberOfJobsExecuted;
```

متد `GetMetaData()`، اطلاعاتی در مورد مدیر وظایف می‌دهد. نوع برگشتی این متد، `SchedulerMetaData` است. یکی از خصیصه‌های این نوع، `NumberOfJobsExecuted` نام دارد که تعداد دفعات اجرای تمامی وظایف تا زمان حال را برگشت می‌دهد.

7) سوال: چگونه می‌توان زمان آغاز به کار مدیر زمانبندی را متوجه شد؟

پاسخ: یکی دیگر از خصیصه‌های نوع `SchedulerMetaData`، `RunningSince` نام دارد که بدین منظور استفاده می‌شود.

```
SchedulerMetaData metaData = scheduler.GetMetaData();  
DateTimeOffset? runningSince = metaData.RunningSince;
```

ادامه دارد...

نظرات خوانندگان

نویسنده: رضا

تاریخ: ۱۰:۳۸ ۱۳۹۱/۰۵/۲۷

آقای راد خیلی ممنون از مطلب مفیدتون. اگر بخواهیم مثلاً یک قسمت از UI رو هر چند دقیقه آپدیت کنیم، نحوه انجام ای کار به چه صورتی خواهد بود؟

نویسنده: بهروز راد

تاریخ: ۱۳:۱۹ ۱۳۹۱/۰۵/۲۷

البته نگفتید که منظورتون desktop یا وب هست، با این فرض که در مورد desktop می‌پرسید، یکی از روش‌ها اینه که یک متغیر عمومی تعریف کنید که ارجاعی به فرمی که قرار هست آپدیت باشه داشته باشه:

```
using System.Windows.Forms;

public static class GlobalData
{
    public static Form ScheduleForm { get; set; }
}
```

در سازنده‌ی فرم می‌تونید اون رو به فرم جاری مقداردهی کنید:

```
GlobalData.ScheduleForm = this;
```

با این فرض که قرار هست عنوان یک دکمه در فرم با نام myButton به My Text تغییر کنه، کلاس پیاده ساز اینترفیس IJob به صورت زیر خواهد بود.

```
namespace SchedulerDemo.Jobs
{
    using System.Linq;
    using System.Windows.Forms;
    using Quartz;

    public class HelloJob : IJob
    {
        private delegate void ButtonTextWriter(string buttonId, string text);
        MainForm form = GlobalData.ScheduleForm as MainForm;

        private void SetButtonText(string buttonId, string text)
        {
            (form.Controls.Find(buttonId, true).FirstOrDefault() as Button).Text = text;
        }

        public void Execute(IJobExecutionContext context)
        {
            form.BeginInvoke(new ButtonTextWriter(SetButtonText), new object[] { "myButton", "My Text"
        });
    }
}
```

نویسنده: امیرحسین جلوداری

تاریخ: ۲۰:۱ ۱۳۹۱/۰۶/۱۳

سلام ... اگر دیده باشید سرویس‌های بلاگ نویسی سرویسی دارن که تو اون میشه یه پستو تایم ارسال بهش داد و در اون روز و ساعت پست تو وبلاگ نمایش داده میشه ... حالا آیا با Quartz.net میشه این کارو کرد؟ (تو asp.net mvc) ... اگه بشه فکر میکنم اگه راجع به این قضیه پستی بزنین خیلی خوب میشه: دی ... ممنون ...

نویسنده: بهروز راد
تاریخ: ۲۰:۱۱ ۱۳۹۱/۰۶/۱۳

این کار ارتباطی با کتابخانه‌های زمانبندی ندارد. اون‌ها پست‌ها رو بر اساس تاریخ و ساعت نشون میدن => پست هایی که تاریخ نمایش اون‌ها از زمان جاری قدیمی‌تر هست نشون داده میشن و پست هایی که تاریخ نمایش اون‌ها از تاریخ جاری جدیدتر هست نشون داده نمیشن.

نویسنده: امیرحسین جلوداری
تاریخ: ۰:۳۷ ۱۳۹۱/۰۶/۱۴

در تاریخ و زمان مشخصی نمایش داده میشن بدون اینکه ادمین وبلاگ به وبلاگش سر بزنه پست در اون زمان مشخص شده در ویلاگ قرار میگیره و همگان اونو میبینن! ... این زمان بندی نیست الان؟! ... یا مثال دیگه : سیستم‌های یادآور یا ToDo که با توجه به زمان و ایمیلی که مشخص کردین براتون میلو ارسال میکنن (اینا منظورم بود: دی)

نویسنده: وحید نصیری
تاریخ: ۰:۵۲ ۱۳۹۱/۰۶/۱۴

همانطور که عنوان شد طراحی دیتابیس است نه استفاده از ابزارهای جانبی. یک وبلاگ در این حالت شبیه به کوئری زیر کار می‌کند (هر فراخوانی صفحه‌ای معادل است با یک کوئری از بانک اطلاعاتی):

```
select * from tblPosts where (showDate is null) or (showDate<=getdate())
```

فیلد showDate اگر نال بود، یعنی یک مطلب معمولی است که درجا نمایش پیدا می‌کند. اگر تاریخی برای آن مشخص شده بود، بر اساس تاریخ جاری یک مقایسه صورت گرفته و رکوردها انتخاب و نمایش داده می‌شوند.

نویسنده: امیرحسین جلوداری
تاریخ: ۰:۵۸ ۱۳۹۱/۰۶/۱۴

ممنون ... ولی بنده منظورم این نبود که الگوریتمش چجوریه! ... منظورم این بود که چیکار کنیم که پشت پرده و بدون این که توسط کاربر Request به پیجی داشته باشیم این کار انجام بشه! ... تو Application_start باید این کارو کرد؟! ... اگه آره چجوری؟

نویسنده: وحید نصیری
تاریخ: ۱:۴ ۱۳۹۱/۰۶/۱۴

سؤال شما این بود: «... تو وبلاگ نمایش داده میشه ...» این یعنی ارسال یک کوئری به بانک اطلاعاتی و دریافت پاسخ. و چرا پشت پرده باید انجام بشه؟ یک وبلاگ بر اساس درخواست یک کاربر هست که صفحاتش نمایش داده میشه. یعنی همون کوئری فوق.

نویسنده: امیرحسین جلوداری
تاریخ: ۱:۱۰ ۱۳۹۱/۰۶/۱۴

کاربر میگه که میخوام 4 ساعت دیگه نمایش بدی! ... یعنی 4 ساعت دیگه باید این Task انجام بشه! ... حالا شما فکر کنین هزار تا کاربر داریم! هر کدوم یه تایمی رو در آینده میدن که پستشون به بازدیدکنندگان ویلاگشون نمایش داده بشه! ... خوب الان یه جورایی ما تو برنامه‌مون یه ترد بینهایت میخوایم که به لیست Taskها در زمان‌های مشخص شده رسیدگی کنه! ... بدون این که دیگه کاربر دخالتی داشته باشه! و همیشه هم این ترد باید کار کنه! ... پس اینجوری باید این ترد تو Application_start تنظیم بشه و تا وقتی سرور روشنه این کارارو انجام بده! ... اگه درست نمیگم تصحیح کنید!

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۱۴ ۱:۱۵

ما داریم در مورد یک وبلاگ بحث می‌کنیم. task نمایش در اینجا یعنی باز کردن یک صفحه توسط کاربر نهایی. در این حالت ابتدا یک کوئری به بانک اطلاعاتی ارسال شده و بر اساس فیلد showDate که پیشتر توسط نویسنده‌ها تنظیم شده (یا خیر ... حالت اختیاری)، اطلاعات نمایش خواهند یافت. نیازی به ترد اضافی نیست. نیازی به ابزار زمانبندی نیست. فقط یک کوئری ساده روی فیلد showDate یک رکورد موجود است.

نویسنده: mgh
تاریخ: ۱۳۹۱/۰۶/۳۱ ۲۰:۱۶

دوست عزیز برنامه‌های زمان بندی ابزاری بسیار زیبا و مفید هستند در صورتی که در مکان و زمان مناسب از آنها بهره ببریم. جهت انجام این کار بسیار ساده ای که مد نظر شماست، ایجاد یک Task به مانند پیچاندن لقمه دور سر است. این کاری که شما مثال زدید رو جناب نصیری خیلی روان و ساده توضیح دادند. هیچ کار اضافی هم نیاز نیست. تنها یک کوئری هست
موفق باشید (:

نویسنده: محمد فخرآوری
تاریخ: ۱۳۹۱/۰۹/۰۵ ۱:۱۴

با سلام
ممنون بابت راهنمایی‌های آقای راد.
آقای راد در مورد پست اول خودتون اومدید یه کلاس نوشتید و در پست دوم در مورد dll توضیح دادین.
اگر ممکن یه سمل بزارین

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۹/۰۵ ۱۰:۲۴

در پست اول، گام به گام، کدهای یک مثال توضیح داده شده. شما فقط Copy & Paste کنید.

نویسنده: حسین
تاریخ: ۱۳۹۱/۰۹/۰۵ ۱۹:۵۳

ممنون از مقالاتون

فقط یه سوالی که برام پیش اومده ،اینه که ایا این کتابخانه از چند نخ برای اجرای کارها استفاده می‌کنه ؟

نویسنده: بهروز راد
تاریخ: ۱۳۹۱/۰۹/۰۶ ۷:۱۱

این کتابخانه، متن باز هست. می‌تونید کدهاش رو بررسی کنید.

نویسنده: محمد فخرآوری
تاریخ: ۱۳۹۱/۱۰/۱۶ ۱۷:۲۰

با سلام
اگر میشه سмпلی بزارید .

نویسنده: حسین

تاریخ: ۹:۴۱ ۱۳۹۱/۱۰/۲۹

سلام.

من توی برنامه از این کتابخانه استفاده کردم و به صورت تستی گذاشتم که هر یک دقیقه یک کار خاصی رو انجام بده. ولی یک مشکلی که دارم اینه که وقتی برنامه رو می بندم به طور کامل از برنامه خارج نمیشه و هنوز پشت صحنه عمل مربوطه رو انجام میده و بایستی دکمه Stop رو بزنم تا اجراش متوقف بشه.

میخواستم بدونم راه گرفتن یک Scheduler خاص چیست و چطور باید Shutdown اش کرد؟ ممنون.

نویسنده: بهروز راد

تاریخ: ۱۰:۵۱ ۱۳۹۱/۱۰/۲۹

چون Threadی که ایجاد میشه، اصطلاحاً Foreground هست و باید به Background تبدیل بشه.

[جزئیات بیشتر...](#)

در ضمن توجه داشته باشید که اگر در وسط اجرای یک فرایند اون رو به زور متوقف کنید، مشکل هرز رفتن منابع رو خواهید داشت. در نظر بگیرید که فایلی رو باز کردید و قصد نوشتن در اون رو دارید، در همین هنگام پروسه رو قطع می کنید. فایل باز می مونه و پروسس های دیگه نمی تونن به اون دسترسی داشته باشن. به نظر من بهتره رفتار پیش فرض Quartz.NET رو تغییر ندید و بر روی منطق برنامه ی خودتون معطوف باشید.

نویسنده: M.Q

تاریخ: ۲۲:۷ ۱۳۹۱/۱۱/۱۷

با سلام؛ صرف نظر از اینکه ما کدهای زمانبندی رو چطور می نویسیم یا از چه کتابخانه ها و روش هایی استفاده می کنیم، برای پیاده سازی برنامه ای هیچ قسمت یوزر اینترفیس ای ندارد (همانند وبلاگ ها یا برنامه های تیکتینگ) و صرفا اطلاعاتی را از دیتابیس (موجود) فراخوانی و عملیاتی روی آنها انجام می دهد، (مثلا ارسال ایمیل یا پیامک)، از چه نوع application ای استفاده کنیم؟ (webapp - winapp - winservice).

من برای انجام اینکار از winapp استفاده نمی کنم چون نمی خواهم یک برنامه را روی ویندوز اجرا کنم. همچنین از webapp استفاده نمی کنم بنا به دلایلی خاص. در حال حاضر از windows service ها استفاده می کنم. آیا راه حل مناسب تری وجود دارد؟

با تشکر از شما

نویسنده: سعید

تاریخ: ۲۳:۱۳ ۱۳۹۱/۱۱/۱۷

سرور متعلق به خودتون است؟ دسترسی بالایی دارید مثل دسترسی نصب و همچنین دسترسی اجرای برنامه به صورت سرویس؟ اگر بله، استفاده از سرویس های ویندوز nt روش متداولی است. اگر نه، برنامه های وب با حداقل دسترسی همین کار رو می تونند انجام بدن.

نویسنده: M.Q

تاریخ: ۱۰:۴۶ ۱۳۹۱/۱۱/۱۸

ممنون از شما آقا سعید

بله در بعضی موارد سرور می تواند از خود من باشد و در بعضی موارد دسترسی ریموت برای نصب و کنترل سرویس را دارم.

درمورد برنامه های وب برای این کار : من یکبار این کار را انجام دادم و از **cache expiration** استفاده کردم. ولی چون این

برنامه‌ی وب هیچ request ای نداشت برنامه down شد و عملیات زمان بندی متوقف شد.

پس میشه گفت استفاده از سرویس‌های ویندوزی، راه حل متداولی هست؟

نویسنده: سعید

تاریخ: ۱۳۹۱/۱۱/۱۸ ۱۰:۵۳

- تولید درخواست به یک سایت کار سختی نیست. از این سرویس‌هایی که مدام سایت رو ping می‌کنند زیاد هست. مثل سرویس‌های فید خوان. مثل سایت‌هایی که بررسی می‌کنند سایت شما آپ است یا داون.

- نه الزاما. سرویس task scheduler ویندوز همیشه در حال اجرا است؛ دقیق و منظم. این سرویس می‌تونه خیلی راحت یک فایل exe معمولی یا حتی یک فایل bat رو در زمان‌های مشخصی اجرا کنه. برای کارهای مدیریت سیستم زیاد ارزش استفاده می‌شه.

نویسنده: احمدی

تاریخ: ۱۳۹۲/۰۵/۲۲ ۱۰:۲۹

با سلام؛ در قسمت تبلیغات سایتی کاربر بازه نمایش تبلیغ خود را وارد می‌کنه و من در مدیریت تأیید می‌کنم در زمان تأیید در قسمت تبلیغات نامه ای برای ایشان ارسال می‌کنم که از این تاریخ تبلیغ شما نمایش داده می‌شود مشکل من در اینجاست که می‌خواهم در زمان خاتمه تاریخ تبلیغ هم نامه ای برای شخص ارسال شود و به ایشان اطلاع داده شود که تبلیغ پایان یافته است.

راه حلی که من برای آن در نظر گرفته بودم این است که در زمان تأیید اولیه تبلیغ با کوارتز یک Job درست کنم و با futuredate از وی بخواهم در روز اتمام تبلیغ ایمیل بزند سوال من این است که :

1- آیا فعال بودن یک job آن هم در بازه‌هایی مانند 6 ماه یا بیشتر به تعداد بالا سرعت سرور را کاهش نمی‌دهد؟
2- از آنجا که تبلیغ‌ها با هم متفاوت است و به فراخور آن نامه‌های ارسالی متفاوت است چگونه می‌توانم به ازای هر تبلیغ به صورت داینامیک Job مربوطه را تخصیص دهم یعنی آن که دستی with identity نزنم(هارد کد نزنم)
پیشاپیش سپاسگزارم

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۰۵/۲۲ ۱۰:۴۶

نیازی نیست به ازای هر تبلیغ یک جاب درست کنید. کلا یک جاب درست کنید که مثلا هر 5 دقیقه یکبار از دیتابیس کوئری بگیره. مواردی رو که منقضی شده (مقایسه تاریخ سرور با تاریخ فیلد انقضاء تبلیغ) پیدا کنه و براشون ایمیل ارسال کنه. به این صورت سربرار خیلی خیلی کمی خواهید داشت.

نویسنده: منصوره

تاریخ: ۱۳۹۲/۱۱/۰۲ ۹:۵۴

سلام من تازه برنامه نویسی asp رو شروع کردم. در سایتی که در حال طراحی آن هستم یک مدیر سیستم دارم و یک سری مشتری که هرکدام نام کاربری و پسورد خودشان را دارند. می‌خواهم برنامه ام به گونه ای باشد که هرروز به طور خودکار بررسی کند اگر تولد یک مشتری است به او اس ام اس دهد یا اگر قرارداد مشتری در حال اتمام است به او اس ام اس دهد و به اطلاعاتش برساند. من برای این کار یک برنامه کوارتز نوشته ام ولی مشکلی که دارم این است که نمی‌دونم این برنامه را از کجا صدا بزنم. نمی‌تونم هر مشتری که لاگین می‌شود این برنامه رو صدا بزنم. چون ممکن است در طول روز مشتری چندین دفعه لاگین شود. می‌خواهم به گونه ای باشد که دفعه اول که مدیر سیستم وارد سایت شد همون بار اول جاب فراخوانی شود و تا بی نهایت ادامه یابد. جاب را به گونه ای تنظیم کرده ام که روزی یک بار اجرا می‌شود. اگر جاب را همون دفعه اولی که مدیر سیستم وارد می‌شود

صدا بزنم درست است؟ بعد از اینکه مدیر سیستم صفحه اش را می بندد اجرای برنامه متوقف می شود یا ادامه می یابد؟

نویسنده: محسن خان
تاریخ: ۱۰:۲ ۱۳۹۲/۱۱/۰۲

[در نظرات قسمت اول](#) بهش اشاره شده. این کدها باید در Application_Start فایل global.asax.cs فراخوانی شوند.

نویسنده: منصوره
تاریخ: ۱۱:۱۹ ۱۳۹۲/۱۱/۰۲

با تشکر ولی این فایل ISchedule را نمی شناسد!؟

نویسنده: محسن خان
تاریخ: ۱۱:۲۹ ۱۳۹۲/۱۱/۰۲

دومین قطعه کدی که در مقاله ی قسمت اول معرفی شده تعریف اینترفیس ISchedule است. قطعه کد اول HelloJob هست از بالا و قطعه کد بعدی ISchedule.

نویسنده: منصوره
تاریخ: ۱۱:۳۴ ۱۳۹۲/۱۱/۰۲

من کلاس ISchedule رو تعریف کردم. برای تعریف global روی نام پروژه راست کلیک کردم و add neww item را انتخاب کردم و از آنجا global را انتخاب کرده ام . آیا این روش درست است؟

```
<%@ Application Language="C#" %>
<script runat="server">
void Application_Start(object sender, EventArgs e)
{
}
</script>
```

نویسنده: محسن خان
تاریخ: ۱۱:۴۰ ۱۳۹۲/۱۱/۰۲

همانطور که قبلا عنوان شد، تعریف myTask.Run را در Application_Start قرار بدید.

نویسنده: میثم 99
تاریخ: ۱۹:۳۱ ۱۳۹۲/۱۲/۰۲

با سلام: اگه می توانید یک مثال از Listener بگذارید ممنون می شوم. به این صورت که چک کند اگر کار قبلی در حال اجرا هست دیگر اجرا نشود و اگر کار به اتمام رسیده بود شروع به اجرای تابع مورد نظر کند.

نویسنده: وحید نصیری
تاریخ: ۲۱:۳۹ ۱۳۹۲/۱۲/۰۲

متد [GetCurrentlyExecutingJobs](#) لیست job های موازی در حال اجرا را می دهد:

```
var jobs = scheduler.GetCurrentlyExecutingJobs();
```

از نتیجه ی آن برای تصمیم گیری استفاده کنید؛ مثلا اگر این لیست خالی یا نال بود آنگاه job جاری اجرا شود.

نویسنده: میثم 99
تاریخ: ۲۱:۴۷ ۱۳۹۲/۱۲/۰۲

ممنون

IJobListener رو پیاده کردم. یک متد داره به نام JobWasExecuted در ابتدا یک متغیر استاتیک بولین تعریف کردم. وقتی تابع اجرا میشه یک میشود. و در JobWasExecuted اون متغیر رو 0 میکنم. تو تست خوب جواب داد. کاملاً درست کار میکنه وقتی کار اتمام پیدا کنه اجرا میشه

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۲ ۲۲:۰۰

روش دیگر استفاده از [MergedJobDataMap](#) هست برای اشتراک گذاری داده‌ها در context آن.

نویسنده: میثم 99
تاریخ: ۱۳۹۲/۱۲/۰۲ ۲۲:۰۸

با تابع GetCurrentlyExecutingJobs () هم انجام شد پیاده سازیش هم راحت‌تره ممنون.

نویسنده: منصوره
تاریخ: ۱۳۹۲/۱۲/۱۹ ۱۳:۰۸

سلام من در برنامه برای ارسال اس ام اس از کوارتز استفاده کردم درست هم کار میکنه ولی مشکلی که داره اینه که می‌خواهم روزی یکبار عمل ارسال اس ام اس انجام بشه بنابراین trigger را به صورت زیر تعریف کرده ام.

```
ITrigger trigger =
TriggerBuilder.Create().WithIdentity("trigger1").StartAt(starttime).WithSimpleSchedule(x =>
x.WithIntervalInHours(24).RepeatForever()).Build ();
```

ولی انگار که این کد کار نمی‌کنه چون تقریباً ساعتی دو بار داره اس ام اس رو میفرسته لطفا راهنمایی ام کنید که مشکل چیه ؟ در ضمن ارسال اس ام اس در قسمت application_start به صورت زیر تعریف شده

```
double h = DateTime.Now.Hour;
h += 7.5;
if (h >= 8)
{
SchedulerDemo.Interfaces.ISchedule mytask = new SchedulerDemo.Jobs.Sendschedule();
mytask.run();
}
```

اضافه کردن 7.5 به ساعت سیستم به دلیل این است که مطابق با ساعت ایران شود. با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۹ ۱۳:۳۲

در متد application_start لاگ کنید برنامه در چه زمان‌هایی [ری‌استارت](#) شده.

نویسنده: منصوره
تاریخ: ۱۳۹۲/۱۲/۱۹ ۱۴:۰۶

ببخشید من درست متوجه منظورتون نشدم. لاگ سایت را از سرور گرفتم ولی چیزی ازش متوجه نشدم. آیا منظورتون این لاگه ؟ میشه بیشتر توضیح بدید ممنون

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۹ ۱۴:۰۹

در یک فایل متنی ساده ثبت کنید، متد application_start چندبار در طول روز فراخوانی شده یا حتی در طول یک ساعت (هر

بار که این متد فراخوانی می‌شود، جایی آن را ثبت کنید). استفاده از [ELMAH](#) هم برای اینکار مفید است. ممکن است برنامه بیش از حد ری‌استارت می‌شود.

نویسنده: افشین

تاریخ: ۱۱:۴ ۱۳۹۲/۱۲/۲۰

یه برنامه نوشتم، که کاربر میتونه زمان رو انتخاب و Job رو ایجاد کنه... لیست Job ها هم در CheckBoxList نشون داده میشه تا کاربر هر کدوم رو که دوست داشتم موقتا Pause و Resume کنه.

Job ها بدون نقص کار میکنن منتها Job ها بعد از Pause و Resume دیگه فعال نمیشن. مشکل از کجاست؟

```
public void Run()
{
    ...
    chkJobs.Items.Add(job.Key, true);
}
```

```
private void chkJobs_ItemCheck(object sender, ItemCheckEventArgs e)
{
    int index = e.Index;
    var sch = Scheduler.GetScheduler();
    foreach (object item in chkJobs.Items)
    {
        JobKey key = (JobKey)item;

        if (e.NewValue == CheckState.Unchecked)
        {
            sch.PauseJob(key);
        }
        else
        {
            sch.ResumeJob(key);
        }
    }
}
```

```
public class Jobs : IJob
{
    public void Execute(IJobExecutionContext context)
    {
        JobDataMap datamap = context.JobDetail.JobDataMap;
        switch (Convert.ToInt32(datamap.GetString("OperationType")))
        {
            case 1:
                MessageBox.Show(datamap.GetString("OperationValue"));
                break;
        }
    }
}
```

نویسنده: منصوره

تاریخ: ۹:۵۵ ۱۳۹۲/۱۲/۲۱

سلام من این کارو انجام دادم متد application_start مرتب ری‌استارت میشه تقریبا هر 20 دقیقه یه بار. سرور ما یک هاست اشتراکی است. میشه لطفا راهنمایی کنید که مشکل چی می‌تونه باشه؟

نویسنده: وحید نصیری

تاریخ: ۱۰:۷ ۱۳۹۲/۱۲/۲۱

- نکات مطلب «[چه زمانی یک برنامه‌ی ASP.NET ری‌استارت می‌شود؟](#)» را بررسی کنید.

- همچنین اگر هاست نامبرده تمام سایت‌ها را با یک [Application pool](#) مدیریت می‌کند، کرش یکی از چند ده سایت دیگر می‌تواند سبب ری‌استارت شدن سایت شما هم بشود؛ چون برنامه‌ها از همه ایزوله نشده‌اند. راه حل آن ایجاد یک Application pool مجزا به ازای هر سایت هست (توسط هاست‌دار).

نویسنده: منصوره
تاریخ: ۱۰:۱۶ ۱۳۹۲/۱۲/۲۱

ممنون از جوابتون این اتفاق مرتب می‌افتد یعنی هر 17,18 دقیقه یکبار. اگر مشکل از کرش سایتهای دیگه باشه آیا این طور مرتب برنامه ری‌استارت میشه؟

نویسنده: وحید نصیری
تاریخ: ۱۰:۱۹ ۱۳۹۲/۱۲/۲۱

بله. زمانیکه تمام سایت‌ها با یک Application pool مدیریت می‌شوند، تمام آن‌ها توسط یک وهرله از w3wp.exe اجرا خواهند شد. با تعریف Application pool های مجزا، هر سایت، یک وهرله‌ی مجزا از w3wp.exe را به خود اختصاص خواهد داد. یعنی اگر Task manager سرور را بررسی کنید، به ازای هر سایت، یک w3wp.exe با pid مجزا قابل مشاهده است. به این ترتیب اگر pid=1234 کرش کرد، تاثیری روی pid=4321 نخواهد داشت.

نویسنده: منصوره
تاریخ: ۱۰:۳۸ ۱۳۹۲/۱۲/۲۱

من با سرور تماس گرفتم گفتن که همه سایتها pool جدا دارن. توی مقاله هم که نگاه کردم هیچکدوم از موارد ذکر شده برای سایت ما صدق نمی‌کرد. نمیدونید مشکل دیگه می‌تونه ناشی از چی باشه؟ تشکر

نویسنده: وحید نصیری
تاریخ: ۱۱:۵ ۱۳۹۲/۱۲/۲۱

با تنظیمات Application pool همه کار می‌توان کرد. مثلاً تنظیم کرد سایت هر 20 دقیقه یکبار پس از بیکاری از حافظه خارج شود. یا تنظیم کرد اگر مصرف حافظه‌ی برنامه به یک حد مشخصی رسید، سایت ری‌استارت شود. بررسی آن نیاز به بررسی کدهای شما و تنظیمات سرور دارد.

عنوان:	دریافت زمانبندی شده به روز رسانی‌های آنتی ویروس Symantec به کمک کتابخانه‌های Quartz.NET و Html Agility Pack
نویسنده:	سیروان عقیقی
تاریخ:	۸:۳۰ ۱۳۹۲/۰۴/۰۳
آدرس:	www.dotnettips.info
برچسب‌ها:	Regular expressions, Quartz.NET, Html Agility Pack, Asynchronous Programming

در این رابطه آقای راد در دو قسمت به صورت مختصر و مفید این کتابخانه قدرتمند رو همراه با ارائه چندین مثال کاربردی معرفی کردند:

[قسمت اول](#)

[قسمت دوم](#)

در تکمیل قسمت‌های فوق بنده می‌خواهم مثالی رو در این رابطه براتون بذارم، هدف از ارائه این مثال اتوماتیک سازی یک فرآیند روتین می‌باشد، به این صورت که در جایی که بنده مشغول به کار هستم یک سری لایسنس آنتی ویروس برای کلاینت‌ها در یک شبکه با مقیاس متوسط تهیه گردیده است، حال یک نسخه رایگان نیز برای کاربرانی که قصد دارند آنتی ویروس را برای سیستم شخصی خود نصب کنند نیز موجود می‌باشد که نیاز به آپدیت دارد معمولاً آپدیت‌ها هر چند روز یکبار یا هر هفته در دو نسخه 64 و 32 بیتی ارائه می‌شوند، روال معمول برای دریافت آپدیت مراجعه به سایت و دانلود نسخه‌های مربوطه می‌باشد.

حال توسط کتابخانه قدرتمند [Quartz.NET](#) این فرآیند روتین را به صورت اتوماتیک می‌خواهیم انجام دهیم، استفاده از کتابخانه ذکر شده سخت نیست همانطور که در دو مطلب قبلی مرتبط ذکر گردیده، تنها پیاده سازی چندین اینترفیس است و بس.

```
namespace SymantecUpdateDownloader
{
    using System;
    using System.IO;
    using Quartz;
    using Quartz.Impl;
    using System.Globalization;
    public class TestJob : IJob
    {
        public void Execute(IJobExecutionContext context)
        {
            new Download().Scraping();
        }
    }
    public interface ISchedule
    {
        void Run();
    }
    public class TestSchedule : ISchedule
    {
        public void Run()
        {
            DateTimeOffset startTime = DateBuilder.FutureDate(2, IntervalUnit.Second);

            IJobDetail job = JobBuilder.Create<HelloJob>()
                .WithIdentity("job1")
                .Build();

            ITrigger trigger = TriggerBuilder.Create()
                .WithIdentity("trigger1")
                .StartAt(startTime)
                .WithDailyTimeIntervalSchedule(x =>
                    x.OnEveryDay().StartingDailyAt(new TimeOfDay(7, 0)).WithRepeatCount(0))
                .Build();

            ISchedulerFactory sf = new StdSchedulerFactory();
            IScheduler sc = sf.GetScheduler();
            sc.ScheduleJob(job, trigger);

            sc.Start();
        }
    }
}
```

در این کد که همانند کدهای پیشنهادی مطلب است، در خط 33 از متد `WithDailyTimeIntervalSchedule` استفاده شده است

و همانطور که مشخص است وظیفه تعیین شده و هر روز ساعت 7 اجرا میشود.

مورد بعدی عملیات دانلود فایل می‌باشد که در ادامه مشاهده خواهید کرد، [صفحه ایی](#) که لینک فایل‌های دانلود را ارائه داده است دو نسخه مد نظر ما را در ابتدا لیست کرده است و با استفاده از web scraping می‌توانیم موارد تعیین شده را استخراج کنیم برای این منظور از کتابخانه [htmlagilitypack](#) استفاده میکنیم، تطبیق دو مورد (لینک) اول جهت دریافت نسخه‌های 32 و 64 بیتی به کمک Regular Expression میسر است و همانطور که در شکل زیر مشاهده میکنید از سمت چپ تاریخ به صورت 8 رقم، سه رقم قسمت دوم و ارقام و حروف قسمت سوم است به اضافه پسوند فایل مشخص است :



```
public class Download
{
    static WebClient wc = new WebClient();
    static ManualResetEvent handle = new ManualResetEvent(true);

    private DateTime myDate = new DateTime();
    public void Scraping()
    {
        using (WebClient client = new WebClient())
        {
            client.Encoding = System.Text.Encoding.UTF8;
            var doc = new HtmlAgilityPack.HtmlDocument();
            ArrayList result = new ArrayList();

            doc.LoadHtml(client.DownloadString("https://www.symantec.com/security_response/definitions/download/detail.jsp?gid=savce"));
            var tasks = new List<Task>();
            foreach (var href in doc.DocumentNode.Descendants("a").Select(x =>
                x.Attributes["href"]))
            {
                if (href == null) continue;
                string s = href.Value;
                Match m = Regex.Match(s, @"http://definitions.symantec.com/defs/(\d{8}-\d{3}-v5i(32|64)\.exe)");
                if (m.Success)
                {
                    Match date = Regex.Match(m.Value, @"(\d{4})(\d{2})(\d{2})");
                    Match filename = Regex.Match(m.Value, @"(\d{8}-\d{3}-v5i(32|64)\.exe)");
                    int year = Int32.Parse(date.Groups[0].Value);
                    int month = Int32.Parse(date.Groups[1].Value);
                    int day = Int32.Parse(date.Groups[3].Value);

                    myDate = new DateTime(
                        Int32.Parse(date.Groups[1].Value),
                        Int32.Parse(date.Groups[2].Value),
                        Int32.Parse(date.Groups[3].Value));
                    if (myDate == DateTime.Today)
                    {
                        tasks.Add(DownloadUpdate(m.Value, filename.Value));
                    }
                    else
                    {
                        MessageBox.Show("امروز آپدیت موجود نیست");
                    }
                }
            }
            DownloadTask = Task.WhenAll(tasks);
        }
    }
    private static Task DownloadTask;
    private Task DownloadUpdate(string url, string fileName)
    {
        var wc = new WebClient();
        return wc.DownloadFileTaskAsync(new Uri(url), @"\\10.1.0.15\SymantecUpdate\\" + fileName);
    }
}
```



```
}
```

توضیح کدهای فوق :

ابتدا توسط متد LoadHtml خط 14 صفحه مورد نظر که حاوی لینک‌ها می‌باشد رو Load میکنیم، سپس توسط یک حلقه foreach خط 16 مقدار خصوصیت href تمام لینک‌های موجود در صفحه را استخراج میکنیم مثلا مقدار خصوصیت href در لینک‌ها به صورت زیر می‌باشد :

```
http://definitions.symantec.com/defs/20130622-007-v5i32.exe
```

```
http://definitions.symantec.com/defs/20130622-007-v5i64.exe
```

همانطور که مشخص است در دو مورد فوق تنها نام فایل متفاوت می‌باشد، همانطور که بحث شد برای نام فایل‌ها هم می‌توانیم یک Pattern را به صورت زیر داشته باشیم :

```
(\d{8}-\d{3}-v5i(32|64)\.exe)
```

در خط 20 نیز عملیات تطبیق تمام hrefهای موجود در صفحه را توسط Regular Expression فوق تطبیق می‌دهیم، اگر تطبیق با موفقیت انجام پذیرفت باید نام فایل و همچنین تاریخ موجود در نام فایل را نیز توسط دو Regular Expression استخراج کنیم(خط 23 و 24) در ادامه برای جدا کردن مقادیر سال ، ماه ، روز از امکان Groups در RegEx استفاده کرده ایم:

```
int year = Int32.Parse(date.Groups[0].Value);  
int month = Int32.Parse(date.Groups[1].Value);  
int day = Int32.Parse(date.Groups[3].Value);
```

در ادامه تاریخ استخراج شده را با تاریخ روز جاری مقایسه می‌کنیم اگر مساوی بود عملیات دانلود فایل‌ها توسط یک [Task](#) تعریف شده به صورت [همزمان](#) بر روی سرور مربوطه دانلود می‌شوند. البته لازم به ذکر است که کدهای فوق مسلما نیاز به Refactoring دارند منتها هدف از ارائه این مثال آشنایی بیشتر با کتابخانه‌های فوق می‌باشد.

نکته آخر اینکه برنامه فوق به حالت‌های مختلفی می‌تواند اجرا گردد مثل یک برنامه وب یا یک سرویس ویندوزی و ... ، بهترین حالت یک سرویس ویندوز می‌باشد، ولی در حالت خام در حال حاضر یک ویندوز اپلیکیشن ساده می‌باشد که بر روی سرور RUN شده است که در آینده به صورت یک سرویس ویندوز ارائه خواهد شد.

نظرات خوانندگان

نویسنده: افشین

تاریخ: ۱۳۹۲/۰۴/۱۵ ۸:۱

یه سؤال دارم که همیشه ذهنم رو مشغول کرده
مگه اینترفیس فقط امضا روال‌ها رو نداره؟ پس یک کلاس نیاز داره که بتونه اون متدها رو پیاده سازی کنه و ما ازش استفاده کنیم
غیر از اینه؟
پس در کد زیر

```
IJobDetail job = JobBuilder.Create<HelloJob>()
```

مجبوریم از اینترفیس به عنوان متغیر استفاده کنیم؟

نویسنده: سیروان عقیفی

تاریخ: ۱۳۹۲/۰۴/۱۵ ۱۲:۴۲

بله به همین صورته، این مطلب رو درباره [اینترفیس](#) و این مطلب رو درباره متدهای [Generic](#) بخونید،
متد Create یک متد Generic است که نام کلاسی رو که اینترفیس IJob و Implement کرده را قبول میکند، و در نهایت مقدار بازگشتی این متد از نوع IJobDetail است.

Cron چیست؟

قابلیتی است در سیستم عامل‌های مبتنی بر یونیکس که وظیفه اجرای وظایف در زمانبندی‌های خاص را بر عهده دارد، و به کاربران این امکان را می‌دهد که وظایف را زمانبندی کرده و در دوره‌های مشخص اجرا کنند.

[اطلاعات بیشتر](#)

در حقیقت رشته‌هایی هستند که از هفت قسمت تشکیل شده‌اند که هر قسمت مشخص کننده اعمال مربوط به زمانبندی می‌باشد مثلاً انجام اعمالی :

اجرای وظیفه ایی خاص هر روز ساعت 8 صبح

اجرای وظیفه ایی خاص هر جمعه آخر ماه

و...

این هفت قسمت توسط فاصله از همدیگر جدا می‌شوند :



عبارات cron خیلی قدرتمند هستند و در عین حال مقداری پیچیده، هدف از ارائه این مطلب آشنایی با این نوع از عبارات و استفاده از آنها در Quartz.NET می‌باشد.

قبل از هر چیز ابتدا باید با فرمت این عبارات آشنا شویم :

Field Name	Mandatory	Allowed Values	Allowed Special Characters
Seconds	YES	0-59	, - * /
Minutes	YES	0-59	, - * /
Hours	YES	0-23	, - * /
Day of month	YES	1-31	, - * ? / L W
Month	YES	1-12 or JAN-DEC	, - * /
Day of week	YES	1-7 or SUN-SAT	, - * ? / L #
Year	NO	empty, 1970-2099	, - * /

پس عبارات cron در ساده‌ترین حالت می‌توانند به صورت :

* * * * ? *

و در حالت پیچیده‌تر می‌توانند به صورت زیر باشند :

0/5 14,18,3-39,52 * ? JAN,MAR,SEP MON-FRI 2002-2010

کارکترهای خاص :

* به معنی "تمام حالات" می‌باشد به عنوان مثال در فیلد "دقیقه" به معنی هر دقیقه می‌باشد.

? به معنی "نگذاشتن مقداری خاص" می‌باشد، به عنوان مثال می‌خواهیم کار ما در یک روز خاص از ماه اتفاق بیفتد اما مهم نیست چه روزی از هفته باشد.

- جهت تعیین یک رنج خاص (محدوده ایی خاص) به عنوان مثال "10-12" در فیلد ساعت به معنی "ساعت هایی 10، 11 و 12" می‌باشد.

, جهت تعیین مقادیر اضافی برای مثال "MON,WED,FRI" در فیلد day-of-week به معنای "روزهای دوشنبه، چهارشنبه و جمعه" می‌باشد.

/ جهت اعمالی increment (کاهش) استفاده می‌شود به طور مثال "15/0" در فیلد seconds به معنای "ثانیه‌های 0، 15، 30، 45" می‌باشد، "15/5" نیز به معنای "ثانیه‌های 5، 20، 35، 50" می‌باشد، به صورت ساده‌تر به طور مثال اگر مقدار "15/0" را در فیلد "minutes" قرار دهیم به معنی "هر 15 دقیقه است و از دقیقه 0 آغاز میشود" با مثلا "20/3" به معنی "هر 20 دقیقه می‌باشد و از دقیقه سوم آغاز میشود".

L "آخرین (Last)" همانطور که از شکل بالا مشخص است تنها در فیلدهایی Day of month و Day of week قابل استفاده می‌باشد به طور مثال اگر در فیلد Day of month استفاده شود به معنای "آخرین روز ماه" می‌باشد.

W "روز هفته (Weekday)"

جهت تعیین Xامین روز ماه به طور مثال مقدار "6#3" به "معنای سومین جمعه ماه" می‌باشد، در واقع مقدار 6 روز و مقدار 3 سومین در ماه می‌باشد.

تعدای مثال :

Expression	**Meaning**
0 0 12 * * ?	Fire at 12pm (noon) every day
0 15 10 ? * *	Fire at 10:15am every day
0 15 10 * * ?	Fire at 10:15am every day
0 15 10 * * ? *	Fire at 10:15am every day
0 15 10 * * ? 2005	Fire at 10:15am every day during the year 2005
0 * 14 * * ?	Fire every minute starting at 2pm and ending at 2:59pm, every day
0 0/5 14 * * ?	Fire every 5 minutes starting at 2pm and ending at 2:55pm, every day
0 0/5 14,18 * * ?	Fire every 5 minutes starting at 2pm and ending at 2:55pm, AND fire every 5 minutes starting at 6pm and ending at 6:55pm, every day
0 0-5 14 * * ?	Fire every minute starting at 2pm and ending at 2:05pm, every day
0 10,44 14 ? 3 WED	Fire at 2:10pm and at 2:44pm every Wednesday in the month of March.
0 15 10 ? * MON-FRI	Fire at 10:15am every Monday, Tuesday, Wednesday, Thursday and Friday
0 15 10 15 * ?	Fire at 10:15am on the 15th day of every month
0 15 10 L * ?	Fire at 10:15am on the last day of every month
0 15 10 L-2 * ?	Fire at 10:15am on the 2nd-to-last last day of every month
0 15 10 ? * 6L	Fire at 10:15am on the last Friday of every month
0 15 10 ? * 6L	Fire at 10:15am on the last Friday of every month
0 15 10 ? * 6L 2002-2005	Fire at 10:15am on every last friday of every month during the years 2002, 2003, 2004 and 2005
0 15 10 ? * 6#3	Fire at 10:15am on the third Friday of every month
0 0 12 1/5 * ?	Fire at 12pm (noon) every 5 days every month, starting on the first day of the month.
0 11 11 11 11 ?	Fire every November 11th at 11:11am.

تولید عبارات cron گاهی اوقات به نظر پیچیده می‌آید (به نظر من که اینطور نیست!) اما برای تولید آسان این عبارات می‌توانید از [این سرویس آنلاین](#) استفاده نمایید.

حال یک مثال در این رابطه :

می‌خواهیم یک کار را هر شب ساعت 11 شب برای انجام زمانبندی کنیم:

```
public class HelloSchedule : ISchedule
{
    public void Run()
    {
        IJobDetail job = JobBuilder.Create<HelloJob>()
            .WithIdentity("job1")
            .Build();

        ITrigger trigger = TriggerBuilder.Create()
            .ForJob(job)
            .WithIdentity("trigger1")
            .StartNow()
            .WithCronSchedule("0 0 23 ? * MON-FRI *")
            .Build();

        ISchedulerFactory sf = new StdSchedulerFactory();
        IScheduler sc = sf.GetScheduler();
        sc.ScheduleJob(job, trigger);

        sc.Start();
    }
}
```

کدهای فوق دقیقاً همانند مثالهایی هستند که پیشتر در سایت مشاهده کرده اید با این تفاوت به جای استفاده از متد `WithSimpleSchedule` از متد `WithCronSchedule` استفاده می‌کنیم که پارامتر ورودی آن عبارت `cron` ما می‌باشد.

نظرات خوانندگان

نویسنده:

محسن جمشیدی

تاریخ:

۱۰:۵۲ ۱۳۹۲/۰۴/۰۸

cron امکان جالبی است برای Quartz.Net اما تا جایی که من اطلاع دارم از شمس پشٹیانی نمی‌کند. آیا اخیراً اقدامی در این زمینه صورت گرفته است؟

نویسنده:

سیروان عقیفی

تاریخ:

۱۳:۳۳ ۱۳۹۲/۰۴/۰۸

بله همینطور، من هم اقدامی رو در این رابطه تا به حال ندیده ام.