

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

```
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 دانلودی 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 میخواهم 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؛ لطفاً در این مورد بیشتر توضیح بفرمائید که چطور آن وظیفه را دوباره به اجرا

در بیاورم.

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 سرویس های دیگر که به سایت ping ارسال می کنند، 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 می تونه 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 مطمئن ترین 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 [پالس می](#) هم می تونی 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 پایین تر هست قابل استفاده نیست.
بهتر است سورس کد کتابخانه را دریافت و کامپایل یا اسمبلی های قدیمی تر 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 عصاری
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 واکنشی کرده و سپس در طی یک حلقه، به آنها پیام ارسال کنید.

نویسنده: محبوبه عصاری
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: ۱۳۹۳/۰۴/۱۱ ۱۲:۱۴

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: ۱۳۹۳/۰۵/۰۲ ۱۱:۳

سلام.

من یه وظیفه نوشتم که می‌یاد هر 1 دقیقه 1 بار یه متن رو در Table درج می‌کنه.

```

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:();

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";
    }
}
```