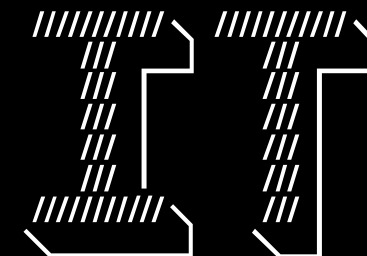


Background processes frameworks in ASP.NET



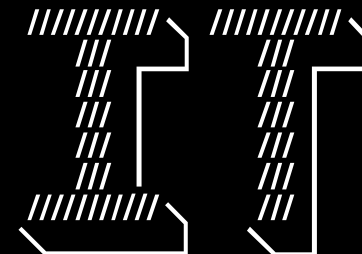
Райффайзен



Зачем это нужно?

- 1) Запускать длительные операции в web сервисах.

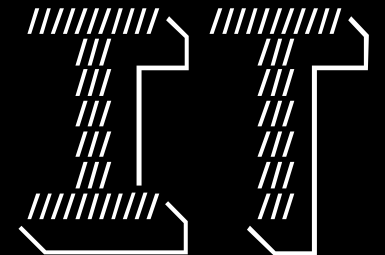
Райффайзен



Зачем это нужно?

- 1) Запускать длительные операции в web сервисах.
- 2) Запускать периодические задачи в web сервисах.

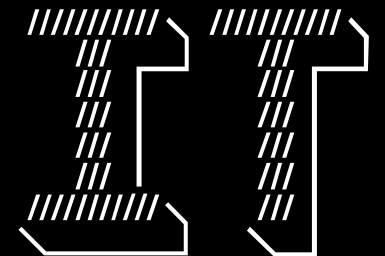
Райффайзен



Зачем это нужно?

- 1) Запускать длительные операции в web сервисах.
- 2) Запускать периодические задачи в web сервисах.
- 3) Присутствует необходимость управлять фоновыми процессами.

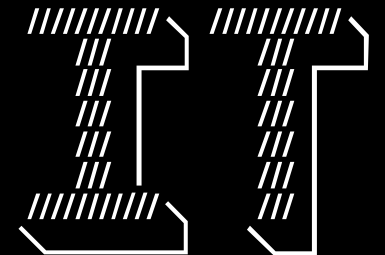
Райффайзен



Кто не рискует...

1) Необработанные исключения.

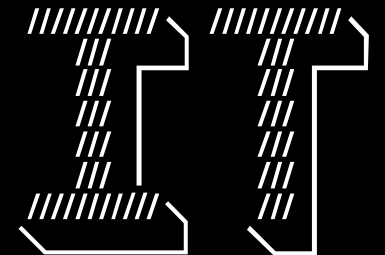
Райффайзен



Кто не рискует...

- 1) Необработанные исключения.
- 2) Линейное масштабирование.

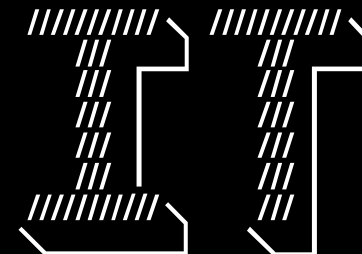
Райффайзен



Кто не рискует...

- 1) Необработанные исключения.
- 2) Линейное масштабирование.
- 3) Прекращение работы AppDomain.

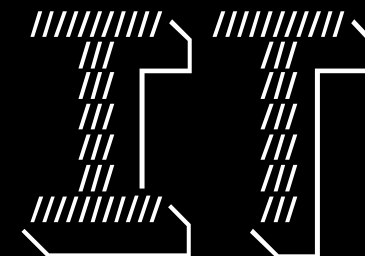
Райффайзен



Кто не рискует...

- 1) Необработанные исключения.
- 2) Линейное масштабирование.
- 3) Прекращение работы AppDomain.
- 4) Производительность системы.

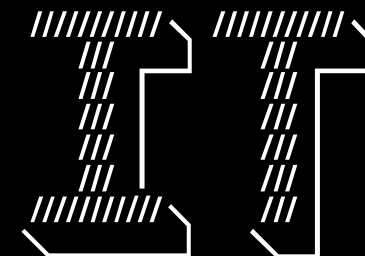
Райффайзен



Кто не рискует...

- 1) Необработанные исключения.
- 2) Линейное масштабирование.
- 3) Прекращение работы AppDomain.
- 4) Производительность системы.
- 5) Инфраструктура.

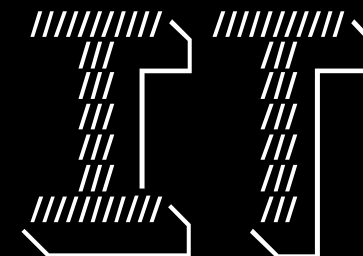
Райффайзен



Определяем требования

1) Какие задачи мы хотим запускать?

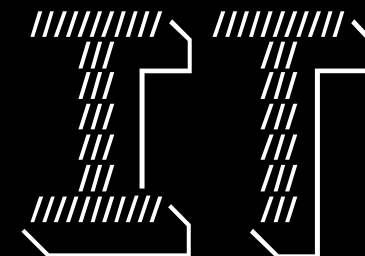
Райффайзен



Определяем требования

- 1) Какие задачи мы хотим запускать?
- 2) Какие настройки должны быть?

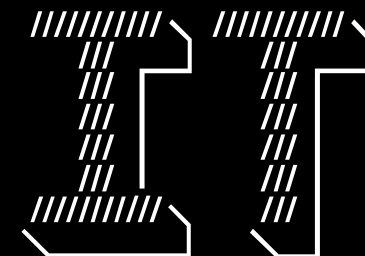
Райффайзен



Определяем требования

- 1) Какие задачи мы хотим запускать?
- 2) Какие настройки должны быть?
- 3) Как будут запускаться задачи?

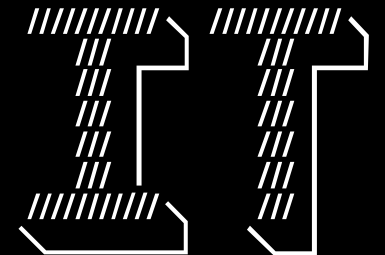
Райффайзен



Определяем требования

- 1) Какие задачи мы хотим запускать?
- 2) Какие настройки должны быть?
- 3) Как будут запускаться задачи?
- 4) Хотим ли мы управлять задачи не только в коде?

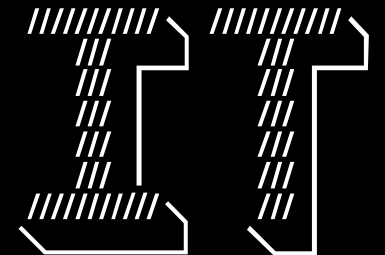
Райффайзен



Определяем требования

- 1) Какие задачи мы хотим запускать?
- 2) Какие настройки должны быть?
- 3) Как будут запускаться задачи?
- 4) Хотим ли мы управлять задачи не только в коде?
- 5) Нужен ли визуальный инструмент?

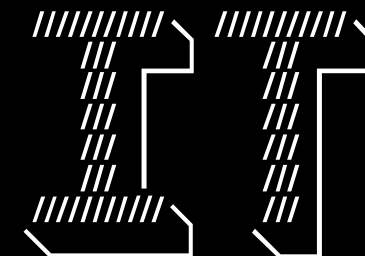
Райффайзен



Пути реализации

1) Сделаю сам!

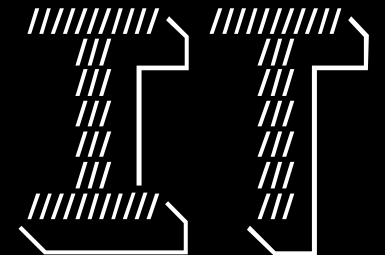
Райффайзен



Пути реализации

- 1) Сделаю сам!
- 2) Сделает кто-то другой.

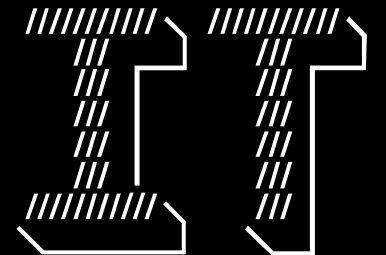
Райффайзен



Пути реализации

- 1) Сделаю сам!
- 2) Сделает кто-то другой.
- 3) Использую, то что мы использовали ранее.

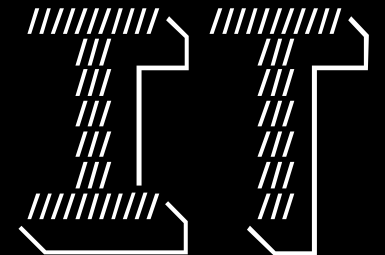
Райффайзен



Пути реализации

- 1) Сделаю сам!
- 2) Сделает кто-то другой.
- 3) Использую, то что мы использовали ранее.
- 4) Найду хороший набор инструментов/Выберу фреймворк.

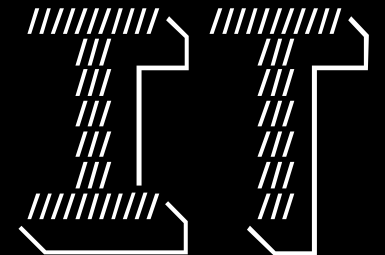
Райффайзен



Что уже есть?

- WebBackgrounder
- QueueBackgroundWorkItem
- FluentScheduler
- Azure
- RecurrentTasks
- Quartz
- Hangfire

Райффайзен

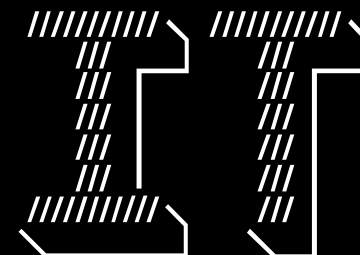


WebBackgrounder

```
public class WebBackgrounderDemo : Job
{
    O references
    public WebBackgrounderDemo(TimeSpan period, TimeSpan delay)
        : base("WebBackgrounder Job", period, delay)
    {
    }

    O references
    public override Task Execute()
    {
        return new Task(() => Console.WriteLine("I am WebBackgrounder"));
    }
}
```

Райффайзен



QueueBackgroundWorkItem

0 references

```
public class QueueBackgroundWorkItem
```

```
{
```

0 references

```
    public void RegisterSomething()
```

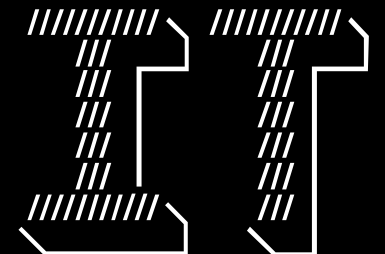
```
    {
```

```
        HostingEnvironment.QueueBackgroundWorkItem(ct => Console.WriteLine("I am WebBackgrounder"));
```

```
    }
```

```
}
```

Райффайзен



RecurrentTasks

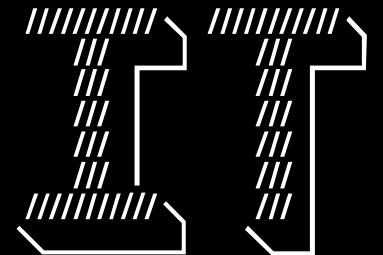
```
0 references
public class RecurrentTasksDemo
{
    0 references
    public void Startup(ServiceCollection services)
    {
        services.AddTask<DemoTask>();
    }
}

3 references
public class DemoTask : IRunnable
{
    private ILogger logger;

    0 references
    public DemoTask(ILogger<DemoTask> logger)
    {
        this.logger = logger;
    }

    0 references
    public Task RunAsync(ITask currentTask, IServiceProvider serviceProvider, CancellationToken cancellationToken)
    {
        return new Task(() => { });
    }
}
```

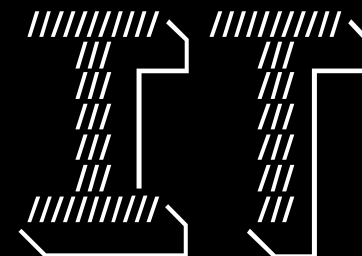
Райффайзен



FluentScheduler

```
Schedule<IJob>().ToRunNow().AndEvery(2).Seconds();  
  
Schedule<DemoJob>().ToRunOnceIn(5).Seconds();  
  
Schedule(() => Console.WriteLine("Hello there")).ToRunEvery(1).Days().At(10, 00);  
  
Schedule<DemoJob2>().ToRunNow().AndEvery(1).Months().OnTheFirst(DayOfWeek.Monday).At(9, 0);  
  
Schedule(() => new DemoJob2()).ToRunNow().AndEvery(2).Seconds();  
  
Schedule<DemoJob3>().AndThen<DemoJob2>().ToRunNow().AndEvery(5).Minutes();
```

Райффайзен

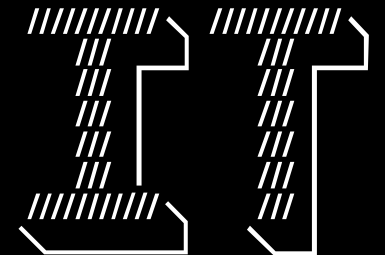


FluentScheduler

Fluent Scheduler Dashboard

#	Name	Next Run Time	State
 	Hello Task	4/15/2019 8:18:57 AM	
 	DoSomething Task	4/15/2019 8:19:17 AM	
 	Basic Task	4/15/2019 8:19:27 AM	

Раиффаизен



Azure

Add WebJob

webapp0909

*

Name

?


myScheduledWebJob

✓

*

File Upload

"ConsoleApp1.zip"



Type

?

Triggered

▼



Scheduled WebJob will be executed based on provided CRON expression. Click here to learn more about CRON Expression.



Triggers

?

Scheduled

▼

*


CRON Expression

?

0 0/20 * * * *

✓

OK





WebJobs


WebJobs provide an easy way to run scripts or programs as background processes.


+

Add

Refresh

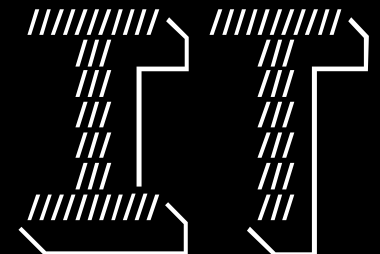
Logs

Delete

Properties

NAME	TYPE	STATUS	SCHEDULE
myScheduledWe...	Triggered	Ready	0 0/20 * * * *
myTriggeredWeb...	Triggered	Ready	n/a
myContinuousW...	Continuous	Pending Restart	n/a

Райффайзен



Quartz

0 references

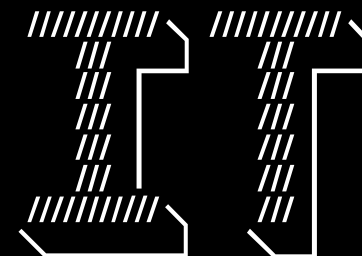
```
public async void Do()
{
    var scheduler = await StdSchedulerFactory.GetDefaultScheduler();
    await scheduler.Start();

    IJobDetail job = JobBuilder.Create<QuartzDemoJob>().Build();

    ITrigger trigger = TriggerBuilder.Create()
        .WithDailyTimeIntervalSchedule
            (s =>
                s.WithIntervalInMinutes(1)
                .OnEveryDay()
                .StartingDailyAt(TimeOfDay.HourAndMinuteOfDay(0, 0))
            )
        .Build();

    await scheduler.ScheduleJob(job, trigger);
}
```

Райффайзен



Quartz

QuartzScheduler

[return to the main site](#)

▶ Start scheduler

⏻ Shutdown

↻ Refresh

Statistics

Running since:	23.04.2017 15:41:37
Total Jobs:	3
Executed Jobs:	0

Properties

Name:	QuartzScheduler
Instance ID:	NON_CLUSTERED
Is remote:	false
Scheduler type:	Quartz.Impl.StdScheduler

DEFAULT

myJob

Trigger	Schedule	Start date	End date	Previous fire date	Next fire date
● myTrigger	repeat every minute	23.04.2017 15:41:29	[none]	[none]	23.04.2017 15:42:29

myJob2

Trigger	Schedule	Start date	End date	Previous fire date	Next fire date
● myTrigger2	repeat 0 times every 3 min	23.04.2017 15:41:29	[none]	[none]	23.04.2017 15:41:29
● myTrigger3	repeat every 2 sec	23.04.2017 15:41:29	[none]	[none]	23.04.2017 15:41:29

⏸ Pause all triggers

▶ Resume all triggers

✕ Delete job

Execute Now

Add Trigger

Hangfire

```
BackgroundJob.Enqueue(() => Console.WriteLine("Fire-and-forget"));

BackgroundJob.Schedule(() => Console.WriteLine("Delayed"), TimeSpan.FromDays(1));

RecurringJob.AddOrUpdate(() => Console.WriteLine("Recurring"), Cron.Daily);

BackgroundJobManager backgroundJobManager = new BackgroundJobManager();

backgroundJobManager.Enqueue(() => Console.WriteLine("Fire-and-forget"));

backgroundJobManager.Schedule(() => Console.WriteLine("Delayed"), TimeSpan.FromDays(1));

RecurringJobManager recurringJobManager = new RecurringJobManager();

RecurringJobOptions options = new RecurringJobOptions() { TimeZone = TimeZoneInfo.Utc };

recurringJobManager.AddOrUpdate("demoJob", Job.FromExpression(() => Console.WriteLine("Recurring")), Cron.Daily(), options);
```

217 / 1

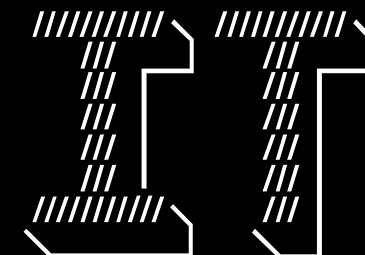
Queues

Queue	Length	Fetches	Next jobs
-------	--------	---------	-----------

DEFAULT 257 20

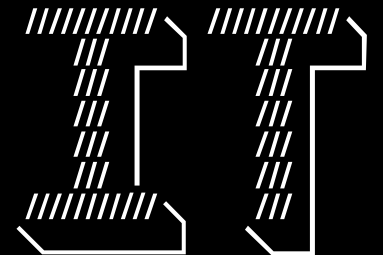
Id	State	Job	Enqueued
07dbeb6f...	Enqueued	IService.SendCampaignEmail	2 minutes ago
0dd0cc32...	Enqueued	IService.SendCampaignEmail	2 minutes ago
9fd2c33d...	Enqueued	IService.SendCampaignEmail	2 minutes ago
450fce40...	Enqueued	IService.SendCampaignEmail	2 minutes ago
5d5ed7fd...	Enqueued	IService.SendCampaignEmail	2 minutes ago

Райффайзен



Спасибо
за внимание!

Райффайзен



Полезные ссылки

- 1) <https://www.hangfire.io/>
- 2) <https://www.quartz-scheduler.net/>
- 3) <https://github.com/fluentcheduler/FluentScheduler>
- 4) <https://github.com/NuGet/WebBackgrounder>
- 5) <https://github.com/justdmitry/RecurrentTasks>

Райффайзен

