

 無料電子ブック

学習

azure-webjobs

Free unaffiliated eBook created from
Stack Overflow contributors.

#azure-
webjobs

.....	1
1: azure-webjobs	2
.....	2
.....	2
Azure WebJobs SDK.....	2
Examples.....	3
AzureWebJob.....	3
2: Azure Webjobs SDK	6
Examples.....	6
JobHost.....	6
.....	6
.....	7
.....	7
.....	7
.....	8
.....	8
Dependency Injection Using Ninject.....	9
.....	11

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [azure-webjobs](#)

It is an unofficial and free azure-webjobs ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official azure-webjobs.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: azure-webjobs をいめる

Azure WebJobは、App Service Webアプリケーション、APIアプリケーション、またはモバイルアプリケーションのコンテキストで、バックグラウンドプロセスとしてスクリプトやプログラムをにするをします。のようなファイルをアップロードしてすることができます。

- .cmd、.bat、.exeWindowsのcmdを
- .ps1 PowerShellを
- .sh bashを
- .php PHPを
- .py Pythonを
- .js Node.jsを
- .jar Javaを

これらのプログラムは、スケジュールcronまたはにWebJobとしてされます。

WebJob SDKをすると、イメージ、キュー、RSS、ファイルのメンテナンス、メールのなど、WebJobができるなタスクのコードをにできます。WebJobs SDKには、Azure StorageおよびService Busをしてをスケジューリングしたり、エラーをしたりするためのがみまれています。

バージョン

Azure WebJobs SDK

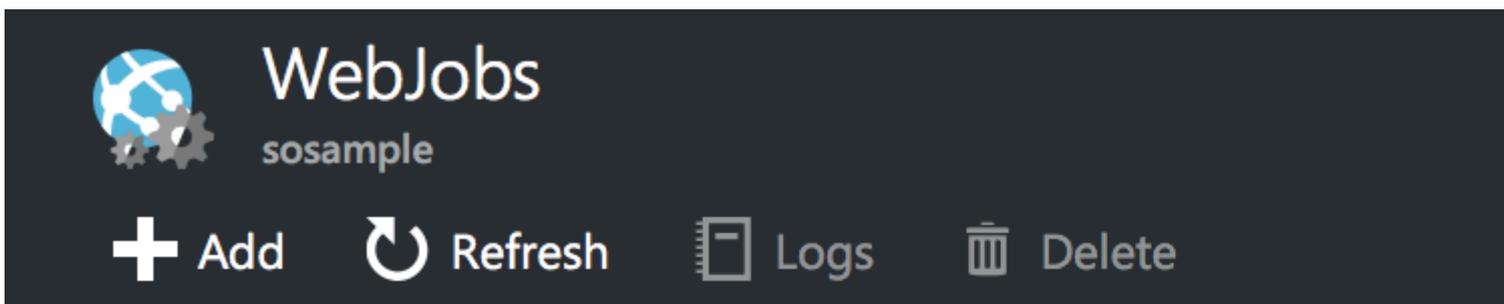
バージョン	
2.0.0-beta1	2016-07-14
1.1.2	2016-04-22
1.1.1	2016-01-13
1.1.0	2015-11-19
1.1.0-rc1	2015-11-02
1.1.0-β1	2015-09-16
1.1.0-alpha2	2015-08-12
1.1.0-alpha1	2015-07-10
1.0.1	2015-03-19
1.0.1-alpha1	2015-02-18

バージョン	
1.0.0	20141017
1.0.0-rc1	2014-09-22
0.6.0-β	2014-09-13
0.5.0-β	2014-09-05
0.4.1-β	2014-08-30
0.4.0-β	2014-08-21

Examples

AzureポータルでのWebJobの

1. [Azure Portal](#)のWebアプリケーションブレードで、[すべての]>[WebJobs]をクリックして、WebJobsブレードをします。



NAME

TYPE

STATUS

TR

You haven't added any WebJobs. Click ADD to get started.

2. []をクリックします。 **Add WebJob**ダイアログがされます。

Add WebJob

sosample

* Name ⓘ

File Upload

Type ⓘ

Scale ⓘ

4.]ボックスで、またはトリガーがcronのスケジュールやウェブフックをすることができます
トリガおみのオプションをします。
トリガおみのオプションをします。
5. [ファイルのアップロード]ボックスで、フォルダアイコンをクリックし、スクリプトがま
れているzipファイルをしします。 zipファイルには、ファイル.exe .cmd .bat .sh .php .py .jsと、
プログラムまたはスクリプトのになサポートファイルがまれているがあります。
6. をチェックして、スクリプトをWebアプリにアップロードします。 WebJobにしたが、
WebJobsブレードのリストにされます。

オンラインで[azure-webjobs](https://riptutorial.com/ja/azure-webjobs/topic/1311/azure-webjobs)をいめるをむ [https://riptutorial.com/ja/azure-webjobs](https://riptutorial.com/ja/azure-webjobs/topic/1311/azure-webjobs)をいめる

2: Azure Webjobs SDK

Examples

JobHost

Azure Webjobs SDKは、トリガーによってされるをし、のAzureサービスAzureストレージやサービスバスなどへのバインディングをにするためのNugetパッケージとしてされるフレームワークです。

SDKは**JobHost**をして、コードされたをします。なシナリオでは、WebjobはJobHostをこのようにするコンソールアプリケーションです。

```
class Program
{
    static void Main()
    {
        JobHostConfiguration config = new JobHostConfiguration();
        config.StorageConnectionString = "Your_Azure_Storage_ConnectionString";
        config.DashboardConnectionString = "Your_Azure_Storage_ConnectionString";
        JobHost host = new JobHost(config);
        host.RunAndBlock();
    }
}
```

JobHostConfigurationをすると、さまざまなトリガーのをパーソナライズできます。

```
config.Queues.BatchSize = 8;
config.Queues.MaxDequeueCount = 4;
config.Queues.MaxPollingInterval = TimeSpan.FromSeconds(15);
config.JobActivator = new MyCustomJobActivator();
```

キューのトリガー

キューメッセージによってトリガされるをするな

```
public static void StringMessage([QueueTrigger("my_queue")] string plainText)
{
    //...
}
```

POCOのシリアライズもサポートしています

```
public static void POCOMessage([QueueTrigger("my_queue")] MyPOCOClass aMessage)
{
    //...
}
```

ブロブのトリガ

AzureストレージBLOBがされたときにトリガされるのな

```
public static async Task BlobTrigger(
    [BlobTrigger("my_container/{name}.{ext}")] Stream input,
    string name,
    string ext)
{
    //Blob with name {name} and extension {ext}

    using (StreamReader reader = new StreamReader(input))
    {
        //Read the blob content
        string blobContent = await reader.ReadToEndAsync();
    }
}
```

ごとにトリガー

SDKは、**6つのフィールド** {second} {minute} {hour} {day} {month} {day of the week} つ **CRON** に
ついてトリガーされたをサポートします。 `JobHostConfiguration` の `UseTimers` が `JobHostConfiguration` 。

```
config.UseTimers();
```

トリガはこのにします

```
// Runs once every 5 minutes
public static void CronJob([TimerTrigger("0 */5 * * * *")] TimerInfo timer)
{
}

// Runs immediately on startup, then every two hours thereafter
public static void StartupJob([TimerTrigger("0 0 */2 * * *", RunOnStartup = true)] TimerInfo
timerInfo)
{
}
```

エラーによるトリガー

エラーはにです。トリガされたの1つでエラーがしたときにトリガされるをできます。

```
//Fires when 10 errors occur in the last 30 minutes (sliding)
public static void ErrorMonitor([ErrorTrigger("0:30:00", 10)] TraceFilter filter)
{
    // get the last 5 errors
    filter.GetDetailedMessage(10);
}
```

エラーのみにちます。

ErrorTriggerはJobHostConfigurationにがです

```
config.UseCore();
```

NuGetパッケージMicrosoft.Azure.WebJobs.Extensionsもインストールするがあります。

スケーリング

Azure WebjobsはAzure Appサービスでします。アプリケーションサービスをにしいインスタンスをすると、インスタンスにのJobHostがされます。

これは、モードでされているWebJobにのみされます。オンデマンドとスケジュールされたWebJobはのスケーリングのをけず、に1つのインスタンスをします。

なWebJobキューメッセージがあり、App Service Planを3インスタンスにすると、のWebJobのインスタンスが3つになります。

に1つのパイプラインがすることをあるがあるため、のインスタンスであるWebJobがするがあります。これらのWebJobの、Singletonをできます。

```
[Singleton]
public static void SingletonQueueProcessing([QueueTrigger("my_queue")] MyPOCOClass aMessage)
{
    //...
}
```

これはAzure Blob Leasesでロックをします。

ログの

WebJobsのダッシュボードには、WebJobのページとのWebJobびしのページの2つのにログがされます。

またはMain()メソッドでびずConsoleメソッドからののは、のメソッドびしのページではなく、WebJobのダッシュボードページにされます。メソッドシグネチャのパラメータからしたTextWriterオブジェクトからののは、メソッドびしの[ダッシュボード]ページにされます。

アプリケーショントレースログをきむには、Console.Out INFOとマークされたログをおよびConsole.Error ERRORとマークされたログををします。

```
public static void WriteLog([QueueTrigger("logqueue")] string message, TextWriter logger)
{
    Console.WriteLine("Console.Write - " + message);
    Console.Out.WriteLine("Console.Out - " + message);
    Console.Error.WriteLine("Console.Error - " + message);
    logger.WriteLine("TextWriter - " + message);
}
```

```
}
```

WebJobのダッシュボードにこれらのメッセージがされます。

```
[07/28/2016 22:29:18 > 0a1c35: INFO] Console.Write - Hello world!  
[07/28/2016 22:29:18 > 0a1c35: INFO] Console.Out - Hello world!  
[07/28/2016 22:29:18 > 0a1c35: ERR ] Console.Error - Hello world!
```

そしてこのメソッドのダッシュボードページにあるこのメッセージ

```
TextWriter - Hello world!
```

Dependency Injection Using Ninject

のは、NinjectをIoCコンテナとしてしてDependency Injectionをするをしています。

まずWebJobプロジェクトにCustomModuleクラスをし、そこにバインディングをします。

```
public class CustomModule : NinjectModule  
{  
    public override void Load()  
    {  
        Bind<IMyInterface>().To<MyService>();  
    }  
}
```

に、JobActivatorクラスをします。

```
class JobActivator : IJobActivator  
{  
    private readonly IKernel _container;  
    public JobActivator(IKernel container)  
    {  
        _container = container;  
    }  
  
    public T CreateInstance<T>()  
    {  
        return _container.Get<T>();  
    }  
}
```

ProgramクラスのMainでJobHostをするときは、JobHostConfigurationにJobActivatorをしてください

```
public class Program  
{  
    private static void Main(string[] args)  
    {  
        //Set up DI  
        var module = new CustomModule();  
        var kernel = new StandardKernel(module);
```

```
//Configure JobHost
var storageConnectionString = "connection_string_goes_here";
var config = new JobHostConfiguration(storageConnectionString) { JobActivator = new
JobActivator(kernel) };

//Pass configuration to JobHost
var host = new JobHost(config);

// The following code ensures that the WebJob will be running continuously
host.RunAndBlock();
}
}
```

に、Functions.csクラスでサービスをします。

```
public class Functions
{
    private readonly IMyInterface _myService;

    public Functions(IMyInterface myService)
    {
        _myService = myService;
    }

    public void ProcessItem([QueueTrigger("queue_name")] string item)
    {
        _myService .Process(item);
    }
}
```

オンラインでAzure Webjobs SDKをむ <https://riptutorial.com/ja/azure-webjobs/topic/2662/azure-webjobs-sdk>

クレジット

S. No		Contributors
1	azure-webjobsをいめる	Community , gbellmann
2	Azure Webjobs SDK	gbellmann , juunas , lopezbertoni , Matias Quaranta