Azure Functions timer trigger

02/27/2017 • 3 minutes to read • Contributors all

In this article

Timer

This article explains how to configure and code timer triggers in Azure Functions. Azure Functions has a timer trigger binding that lets you run your function code based on a defined schedule.

The timer trigger supports multi-instance scale-out. A single instance of a particular timer function is run across all instances.

This is reference information for Azure Functions developers. If you're new to Azure Functions, start with the following resources:

- Create your first function.
- Azure Functions developer reference.
- C#, F#, Node, or Java developer reference.
- Azure Functions triggers and bindings concepts.

Timer trigger

The timer trigger to a function uses the following JSON object in the `bindings` array of function.json:

JSON

```json
{
    "schedule": "<CRON expression - see below>",
    "name": "<Name of trigger parameter in function signature>",
    "type": "timerTrigger",
    "direction": "in"
}
```

The value of `schedule` is a CRON expression that includes these six fields:

Copy

```
{second} {minute} {hour} {day} {month} {day-of-week}
```

ⓘ Note

Many of the cron expressions you find online omit the `{second}` field. If you copy from one of them, you need to adjust for the extra `{second}` field. For specific examples, see Schedule examples below.

The default time zone used with the CRON expressions is Coordinated Universal Time (UTC). To have your CRON expression based on another time zone, create a new app setting for your function app named `WEBSITE_TIME_ZONE`. Set the value to the name of the desired time zone as shown in the Microsoft Time Zone Index.

For example, *Eastern Standard Time* is UTC-05:00. To have your timer trigger fire at 10:00 AM EST every day, use the following CRON expression that accounts for UTC time zone:

JSON Copy

```json
"schedule": "0 0 15 * * *",
```

Alternatively, you could add a new app setting for your function app named `WEBSITE_TIME_ZONE` and set the value to **Eastern Standard Time**. Then the following CRON expression could be used for 10:00 AM EST:

JSON Copy

```json
"schedule": "0 0 10 * * *",
```

Schedule examples

Here are some samples of CRON expressions you can use for the `schedule` property.

To trigger once every five minutes:

JSON Copy

```json
"schedule": "0 */5 * * * *"
```

To trigger once at the top of every hour:

JSON Copy

```json
"schedule": "0 0 * * * *",
```

To trigger once every two hours:

JSON Copy

```json
"schedule": "0 0 */2 * * *",
```

To trigger once every hour from 9 AM to 5 PM:

JSON Copy

```json
"schedule": "0 0 9-17 * * *",
```

To trigger At 9:30 AM every day:

JSON Copy

```json
"schedule": "0 30 9 * * *",
```

To trigger At 9:30 AM every weekday:

JSON Copy

```json
"schedule": "0 30 9 * * 1-5",
```

Trigger usage

When a timer trigger function is invoked, the timer object is passed into the function. The following JSON is an example representation of the timer object.

JSON Copy

```json
{
    "Schedule":{
    },
    "ScheduleStatus": {
        "Last":"2016-10-04T10:15:00.012699+00:00",
        "Next":"2016-10-04T10:20:00+00:00"
    },
    "IsPastDue":false
}
```

Trigger sample

Suppose you have the following timer trigger in the `bindings` array of function.json:

JSON Copy

```
{
    "schedule": "0 */5 * * * *",
    "name": "myTimer",
    "type": "timerTrigger",
    "direction": "in"
}
```

See the language-specific sample that reads the timer object to see whether it's running late.

- C#
- F#
- Node.js

Trigger sample in C#

C# Copy

```
public static void Run(TimerInfo myTimer, TraceWriter log)
{
    if(myTimer.IsPastDue)
    {
        log.Info("Timer is running late!");
    }
    log.Info($"C# Timer trigger function executed at: {DateTime.Now}" );
}
```

Trigger sample in F#

F# Copy

```
let Run(myTimer: TimerInfo, log: TraceWriter ) =
    if (myTimer.IsPastDue) then
        log.Info("F# function is running late.")
    let now = DateTime.Now.ToLongTimeString()
    log.Info(sprintf "F# function executed at %s!" now)
```

Trigger sample in Node.js

JavaScript Copy

```javascript
module.exports = function (context, myTimer) {
    var timeStamp = new Date().toISOString();

    if(myTimer.isPastDue)
    {
        context.log('Node.js is running late!');
    }
    context.log('Node.js timer trigger function ran!', timeStamp);

    context.done();
};
```

Next steps

For information about other bindings and triggers for Azure Functions, see Azure Functions triggers and bindings developer reference.