Configuring Precompiled Functions to Work with Azure Storage

Jason Roberts
.NET MVP

@robertsjason dontcodetired.com

Overview

Azure Storage overview

Introduction to the Azure Storage Emulator

Configuring triggers and bindings:

- Blob storage
- Queue storage
- Table storage

Add Azure Queue storage output bindings

Local testing using the Azure Storage Emulator

Testing using an Azure Storage account

Create Azure Queue and blob-triggered functions

“Microsoft Azure Storage In-depth”

by Alan Smith

An Overview of Azure Storage

Blob storage

Similar to files

Word docs, log files, photos, etc.

Stored in blob containers (like folders)

Access stored blobs via URL, REST, programmatically

Queue storage

Store and retrieve messages

<64KB size

Create processing pipelines

Each function can scale independently

An Overview of Azure Storage

Table storage

Structured NoSQL data store

Schemaless design / flexibility

E.g. Device info, address books, metadata, etc.

Azure Cosmos DB Table API (preview)

File storage & disk storage

Network shares

Disk storage for VMs

Introducing the Azure Storage Emulator

Local machine emulation for

- Azure Blob services
- Azure Queue services
- Azure Table services

Develop and test locally without needing a network connection

No Azure Storage usage costs

Installation

- As part of Azure SDK
- Standalone installer

Introducing the Azure Storage Emulator

Emulator limitations

- Not all operations / features supported
- Blob size max 2 GB
- Total table row size limited <1 MB
- Other potentially important differences depending on what capabilities your app is making use of

Documentation and limitations

https://docs.microsoft.com/en-us/azure/storage/common/storage-use-emulator

http://bit.ly/storageemulator

Configuring Azure Blob Storage Triggers

```
public static void Run(
    [BlobTrigger("my-blob-container/{name}")]Stream myBlob,
    string name,
    TraceWriter log) { … }
```

Binding tokens

{blobname}
{blobextension}

```
[BlobTrigger("samples-workitems/{name}", Connection="...")]
"AzureWebJobsStorage"
```

Blob Storage Trigger Filters

```
[BlobTrigger("my-blob-container/{name}.jpg")]
kittens.jpg
string name == "kittens.jpg"

[BlobTrigger("my-blob-container/original-{name}")]
original-kittens.jpg
string name == "original-kittens.jpg"
```

Configuring Azure Blob Storage Bindings

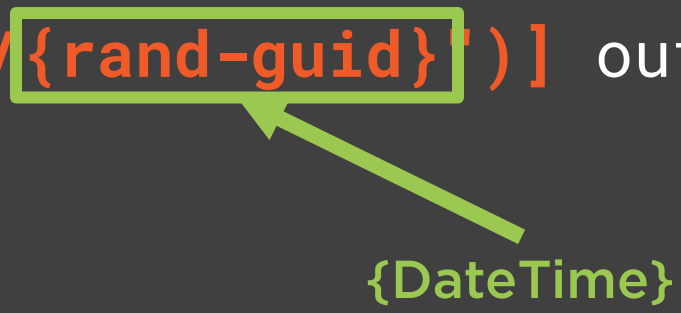
```
public static void Run(

...

    [Blob("accepted-applications/{rand-guid}")] out string
    acceptedApplication,

    TraceWriter log)

{ ... }
```




```
[Blob("accepted-applications/{rand-guid}", Connection="…")]
```

Configuring Azure Queue Storage Triggers

```
public static void Run(
    [QueueTrigger("loan-applications")]
        LoanApplication application,
    [Blob("accepted-applications/{rand-guid}")]
        out string acceptedApplication,
    [Blob("declined-applications/{rand-guid}")]
        out string declinedApplication,
    TraceWriter log) { … }
```

```
[QueueTrigger("loan-applications", Connection="…")]
```

Configuring Azure Queue Storage Bindings

```
[Queue("loan-applications")] out string message,
```

```
[Queue("loan-applications", Connection="…")]
```

Configuring Azure Table Storage Bindings

```
public class Application : TableEntity
{
    public Application(string salesChannel,
                       Guid applicationId)
        : base(salesChannel, applicationId.ToString()) {}
    public Application() {}
    public string Name { get; set; }
    public int Age { get; set; }
}
```

Storage Table Output Bindings

```
[Table("LoanApplications")]
IAsyncCollector<Application> applicationTable


await applicationTable.AddAsync(
    new Application("web", Guid.NewGuid())
        {
            Name = "Amrit",
            Age = 42
        }
);
```

Storage Table Output Bindings

```
[Table("LoanApplications")] out Application application


application = new Application("web", Guid.NewGuid())
                    {
                        Name = "Amrit",
                       Age = 42
                    }


[Table("LoanApplications", Connection = "…")]
```

Storage Table Input Bindings

```
public static void Run(
    [QueueTrigger("applications")]
       string queueItem,
    [Table("LoanApplications", // table name
           "web", // partition key
           "{queueTrigger}")] // row key
       Application application)
{
  …
}
```

Summary

Azure Storage and emulator overview

Configuring triggers and bindings:

- [BlobTrigger] [Blob]
- [QueueTrigger] [Queue]
- [Table]

Queue("loan-applications")]

Testing using the Azure Storage Emulator & test Azure Storage account

[QueueTrigger("loan-applications")]

[BlobTrigger("accepted-applications/{name}")]

Next:

Working with Other Triggers and Azure Services