



BlobTrigger

Binding
T ¹
string
TextReader
Stream
ICloudBlob
CloudBlockBlob
CloudPageBlob

QueueTrigger

Binding
T (as JSON)
string
byte[]
CloudQueueMessage

ServiceBusTrigger

Binding
T (as JSON)
string
byte[]
BrokeredMessage

Blobs

Binding	Access ²	Behavior Notes
T ¹	[R]	Null pass-through. Never creates.
out T ³	[W]	Null pass-through. Always creates.
string	[R]	Null when missing. Never creates.
out string	[W]	Creates when non null value returned.
TextReader	[R]	Null when missing. Never creates.
TextWriter	[W]	Not null. Always creates.
Stream	R	Null when missing. Never creates.
Stream	W	Not null. Creates on write.
CloudBlobStream	[W]	Not null. Creates on write.
ICloudBlob		
CloudBlockBlob	[RW]	Not null. Creates a container.
CloudPageBlob		

Queues

Binding	Behavior Notes
ICollection<T>	Not null. Creates when item added.
IAsyncCollector<T>	
out T (as JSON)	Sends "null" for null value. Always creates. Executed last.
out string	
out byte[]	Sends message when non null value returned.
out CloudQueueMessage	Creates queue when message is to be sent. Executed last.
CloudQueue	Not null. Always creates.

Tables

Binding	Behavior Notes
IQueryable<T>	Not null. Empty when missing. Never creates.
ICollection<T>	
IAsyncCollector<T>	Not null. Creates when item added.
T (where T: ITableEntity)	
T (POCO)	Null when missing. Never creates.
CloudTable	Not null. Always creates.

ServiceBus

Binding	Behavior Notes
ICollection<T>	Not null. Creates when item added.
IAsyncCollector<T>	
out T (as JSON)	Sends message with "null" when null returned. Always creates.
out string	
out byte[]	Sends message when non null value returned.
out BrokeredMessage	Creates queue when message is to be sent.

Misc. Bindings

Argument Type	To
CancellationToken	Monitor for cancellation request.
CloudStorageAccount	Get access to Azure Storage account.
IBinder	Bind an arbitrary attribute at runtime.
TextWriter	Send a log message to a console output.
NoAutomaticTrigger	Indicate no automatic triggering.

Binding Path

Use	In	To
%name%	Any	Replace with a value provided by INameResolver.
{name}	Trigger	Define named parameter to capture a value.
	Binder	Replace with a binding parameter value.
{{ or }}	Any	Insert literal { or }.

Binding Parameters

Use in binding path pattern and for binding to function arguments. Values may be populated by BlobTrigger (BT), QueueTrigger (QT), or retrieved from a triggering POCO. Can be bound either to the exact type or to string. If the exact type is string, it can also be bound as any TryParse-able type.

Name	Type	Source	Value
BlobTrigger	string	BT	Triggering blob path
QueueTrigger	string	QT	Triggering message content if valid string
DequeueCount	int	QT	N times a message has been de-queued
ExpirationTime	DateTimeOffset	QT	Time a message expires
Id	string	QT	A message ID
InsertionTime	DateTimeOffset	QT	Time a message was added to the queue
NextVisibleTime	DateTimeOffset	QT	Time a message will next be visible
PopReceipt	string	QT	A message's pop receipt
"Property Name"	Primitive Type	POCO	A top level property of triggering POCO

Configuration

Put connection string values in Web.config and App.config files, or in Azure Website configuration, or define environment variables, or set SDK connection strings in code.

Name	Value
AzureWebJobsStorage	Azure Storage connection string used for reading and writing data.
AzureWebJobsDashboard	Azure Storage connection string used for logging and diagnostics.
AzureWebJobsServiceBus	Azure ServiceBus connection string.

¹ As de-serialized by ICloudBlobStreamBinder<T>. See notes on Stream read binding behavior.

² The default value is in []. If none specified then explicit FileAccess attribute property is required.

³ Serialized by ICloudBlobStreamBinder<T>. See notes on Stream write binding behavior.

1. Blob Operations

1.1. Simple Blob Trigger

Triggered when a new blob is created or updated at the container "input". Binds the second argument to a blob in a container "output" with the same name.

```
public static void BlobCopySimple(
    [BlobTrigger("input/{name}")] TextReader input,
    [Blob("output/{name}")] out string output);
```

1.2. Advanced Blob Trigger

Triggered when a new blob is created or modified. The "name" and "extension" arguments are bound to parameters captured by the triggering blob path pattern. The "blobTrigger" is bound to the triggering blob path. The "output" argument is bound to a write stream associated with a blob in "output" container with the same name.

```
public static async Task BlobTriggerAdvanced(
    [BlobTrigger(@"staging/{name}.{extension}")] CloudBlockBlob blob,
    string name,
    string extension,
    string blobTrigger,
    [Blob(@"output/{name}.{extension}", FileAccess.Write)] Stream output);
```

1.3. Handle Poison Blob

Invoked when a blob triggering message ends up in the poison queue after five unsuccessful retries. The "log" parameter is bound to a text writer associated with a Dashboard output log.

```
public static void PoisonBlobHandler(
    [QueueTrigger("webjobs-blobtrigger-poison")] BlobTriggerPosionMessage
    poisonMessage,
    TextWriter log);
```

2. Queue Operations

2.1. Simple Queue Trigger

Triggered when new message received from "orders" queue. Binds triggering message to an Order object. Binds the second argument to outgoing message in "output" queue.

```
public static void MessageCopySimple(
    [QueueTrigger("mailbox")] CloudQueueMessage message,
    [Queue("output")] out string output);
```

2.2. Property Binding

Shows binding parameters to properties of a queue message. The "name" parameter is bound to the value of the "Name" property in the Order object. The "dequeueCount" parameter is bound to the property value of the actual CloudQueueMessage object.

```
public static void PropertyBinding(
    [QueueTrigger("order")] Order order,
    string name,
    int dequeueCount,
    string queueTrigger,
    [Blob("orders/{OrderId}")] out string orderData,
    TextWriter log);
```

2.3. Handle Poison Message

Invoked when a message ends up in the poison queue. It is used to demonstrate the poison queue message handling. The trigger message is moved into a poison queue after a binding or a function fails 5 times.

```
public static void PoisonMessageHandler(
    [QueueTrigger("orders-poison")] string poisonMessage);
```

3. Table Operations

3.1. Simple Table Update

Triggered on new message in the "orders" queue. Binds "entries" parameter to the "transactions" table.

```
public static void ProcessOrder(
    [QueueTrigger("orders")] Order order,
    [Table("items")] ICollector<OrderItem> items);
```

3.2. Manual Trigger

Not triggered automatically. Binds "logEntries" to the "log" table.

```
[NoAutomaticTrigger]
public static void ProcessLogEntries(
    [Table("log")] IQueryable<LogEntry> logEntries);
```

4. ServiceBus Operations

4.1. Simple ServiceBus Queue Trigger

Triggered when new message received from "incoming" queue. Binds the "message" argument to outgoing message in "outgoing" queue.

```
public static void ServiceBusMessageCopySimple(
    [ServiceBusTrigger("incoming")] string notification,
    [ServiceBus("outgoing")] out string message);
```

4.2. Simple ServiceBus Topic Listener

Topic subscription listener.

```
public static void ServiceBusEventsListener(
    [ServiceBusTrigger("events", "service-1")] BrokeredMessage message);
```

5. Misc. Operations

5.1. Runtime Binder

Shows binding to IBinder for runtime binding.

```
public static void ProcessRequestWithRuntimeBinder(
    [QueueTrigger("requests")] Request request,
    IBinder binder);
```

5.2. Cancellation Token

Shows binding to a cancellation token. Triggered by a queue message received from a queue with a name resolved by INameResolver.

```
public static void ShutdownMonitor(
    [QueueTrigger("%ShutdownQueueName%")] string message,
    CancellationToken cancellationToken);
```