

# How to create multiple output bindings in Azure Functions using — .NET

---

BLOG CREATED BY — PRATHAMESH ZADKHANDE



# Problem Statement :

---

- ❖ There is a need to send data from an Azure Function to multiple services such as Event Hubs, Service Bus, and Cosmos DB.

# Solution/Architecture :

---

One solution to this problem is to use output bindings in the Azure Function to send data to the desired services. Output bindings allow the function to send data to a variety of Azure services and other external sources.

Technical Details and Implementation of Solution:

To implement this solution, you will need to do the following :

## **1. Install the necessary NuGet packages :**

Install-Package Microsoft.Azure.WebJobs.Extensions.EventHubs

Install-Package Microsoft.Azure.WebJobs.Extensions.ServiceBus

Install-Package Microsoft.Azure.WebJobs.Extensions.CosmosDB

## 2. Define the output bindings in your function's 'function.json' file. For example :

```
{
  "bindings": [
    {
      "type": "eventHub",
      "name": "outputEventHub",
      "connection": "EventHubConnectionString",
      "direction": "out",
      "eventHubName": "EventHubName"
    },
    {
      "type": "serviceBus",
      "name": "outputServiceBus",
      "connection": "ServiceBusConnectionString",
      "direction": "out",
      "queueName": "QueueName"
    },
    {
      "type": "cosmosDB",
      "name": "outputCosmosDB",
      "connection": "CosmosDBConnectionString",
      "direction": "out",
      "databaseName": "DatabaseName",
      "collectionName": "CollectionName"
    }
  ]
}
```

```
1  {
2    "bindings": [
3      {
4        "type": "eventHub",
5        "name": "outputEventHub",
6        "connection": "EventHubConnectionString",
7        "direction": "out",
8        "eventHubName": "EventHubName"
9      },
10     {
11       "type": "serviceBus",
12       "name": "outputServiceBus",
13       "connection": "ServiceBusConnectionString",
14       "direction": "out",
15       "queueName": "QueueName"
16     },
17     {
18       "type": "cosmosDB",
19       "name": "outputCosmosDB",
20       "connection": "CosmosDBConnectionString",
21       "direction": "out",
22       "databaseName": "DatabaseName",
23       "collectionName": "CollectionName"
24     }
25   ]
26 }
```

**3. In your function code, you can use the output binding names defined in 'function.json' to send data to the corresponding service. For example:**

```
[FunctionName("MultipleOutputFunction")]
public static void Run(
    [HttpTrigger] string input,
    [EventHub("outputEventHub", Connection = "EventHubConnectionString")] out string outputEventHub,
    [ServiceBus("outputServiceBus", Connection = "ServiceBusConnectionString")] out string
outputServiceBus,
    [CosmosDB("outputCosmosDB", Connection = "CosmosDBConnectionString")] out dynamic
outputCosmosDB,
    ILogger log)
{
    outputEventHub = "Hello, Event Hub!";
    outputServiceBus = "Hello, Service Bus!";
    outputCosmosDB = new { id = "123", message = "Hello, Cosmos DB!" };
}
```

```
[FunctionName("MultipleOutputFunction")]
public static void Run(
    [HttpTrigger] string input,
    [EventHub("outputEventHub", Connection = "EventHubConnectionString")] out
string outputEventHub,
    [ServiceBus("outputServiceBus", Connection = "ServiceBusConnectionString")]
out string outputServiceBus,
    [CosmosDB("outputCosmosDB", Connection = "CosmosDBConnectionString")] out
dynamic outputCosmosDB,
    ILogger log)
{
    outputEventHub = "Hello, Event Hub!";
    outputServiceBus = "Hello, Service Bus!";
    outputCosmosDB = new { id = "123", message = "Hello, Cosmos DB!" };
}
```

# Challenges in Implementing the Solution :

---

- ❖ One challenge in implementing this solution is that it requires familiarity with Azure Functions, output bindings, and the specific services being used (Event Hubs, Service Bus, and Cosmos DB). It may also be necessary to troubleshoot any issues that arise during the implementation process.

# Business Benefit:

---

- ❖ Using output bindings to send data to multiple services allows for greater flexibility and integration with different systems and services. This can help improve efficiency and decision-making within an organization.