



FREE eBook

LEARNING azureservicebus

Free unaffiliated eBook created from
Stack Overflow contributors.

#azureservi
cebus

Table of Contents

About.....	1
Chapter 1: Getting started with azureservicebus.....	2
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Chapter 2: Get started with queues.....	3
Remarks.....	3
Sample code.....	3
Examples.....	3
1. Create a namespace using the Azure portal.....	3
2. Create a queue using the Azure portal.....	3
3. Send messages to the queue.....	4
Create a console application.....	4
Add the Service Bus NuGet package.....	4
Write some code to send a message to the queue.....	4
4. Receive messages from the queue.....	5
Credits.....	7

About

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

It is an unofficial and free azureservicebus 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 azureservicebus.

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

Chapter 1: Getting started with azureservicebus

Remarks

This section provides an overview of what Azure Service Bus is, and why a developer might want to use it.

It should also mention any large subjects within Azure Service Bus, and link out to the related topics. Since the documentation for Azure Service Bus is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting azureservicebus set up or installed.

Read [Getting started with azureservicebus online](#):

<https://riptutorial.com/azureservicebus/topic/1122/getting-started-with-azureservicebus>

Chapter 2: Get started with queues

Remarks

Sample code

More sample code can be found [here](#).

Examples

1. Create a namespace using the Azure portal

1. Log on to the Azure classic portal.
2. In the left navigation pane of the portal, click **Service Bus**.
3. In the lower pane of the portal, click **Create**.
4. In the **Add a new namespace** dialog, enter a namespace name. The system immediately checks to see if the name is available.
5. After making sure the namespace name is available, choose the country or region in which your namespace should be hosted.
6. Leave the other fields in the dialog with their default values (**Messaging** and **Standard Tier**), then click the OK check mark. The system now creates your namespace and enables it. You might have to wait several minutes as the system provisions resources for your account.

###Obtain the credentials

1. In the left navigation pane, click the **Service Bus** node, to display the list of available namespaces:
2. Select the namespace you just created from the list shown:
3. Click **Connection Information**.
4. In the **Access connection information** pane, find the connection string that contains the SAS key and key name.
5. Make a note of the key, or copy it to the clipboard.

2. Create a queue using the Azure portal

1. Log on to the [Azure classic portal](#).

2. In the left navigation pane of the portal, click **Service Bus**.
3. Select the namespace that you would like to create the queue in. In this case, it is "mytestns1."
4. Select **Queues**.
5. Select **New** in the bottom left corner, then select **Quick Create**.
6. Enter the **Queue Name** and ensure the proper namespace is selected.
7. Select **Create a new queue**.

3. Send messages to the queue

In order to send messages to the queue, we will write a C# console application using Visual Studio.

Create a console application

1. Launch Visual Studio and create a new Console application.

Add the Service Bus NuGet package

1. Right click on the newly created project and select **Manage NuGet Packages**.
2. Click the **Browse** tab, then search for "Microsoft Azure Service Bus" and select the **Microsoft Azure Service Bus** item. Click **Install** to complete the installation, then close this dialog box.

Write some code to send a message to the queue

1. Add the following using statement to the top of the Program.cs file.

```
using Microsoft.ServiceBus.Messaging;
```

2. Add the following code to the `Main` method, set the **connectionString** variable as the connection string that was obtained when creating the namespace, and set **queueName** as the queue name that used when creating the queue.

```
var connectionString = "<Your connection string>";  
var queueName = "<Your queue name>";  
  
var client = QueueClient.CreateFromConnectionString(connectionString, queueName);  
var message = new BrokeredMessage("This is a test message!");  
client.Send(message);
```

Here is what your Program.cs should look like.

```

using System;
using Microsoft.ServiceBus.Messaging;

namespace GettingStartedWithQueues
{
    class Program
    {
        static void Main(string[] args)
        {
            var connectionString = "<Your connection string>";
            var queueName = "<Your queue name>";

            var client = QueueClient.CreateFromConnectionString(connectionString,
queueName);
            var message = new BrokeredMessage("This is a test message!");

            client.Send(message);
        }
    }
}

```

3. Run the program, and check the Azure classic portal. Notice that the **Queue Length** value should now be 1.

4. Receive messages from the queue

1. Create a new console application and add a reference to the Service Bus NuGet package, similar to the sending application above.
2. Add the following `using` statement to the top of the Program.cs file.

```
using Microsoft.ServiceBus.Messaging;
```

3. Add the following code to the `Main` method, set the **connectionString** variable as the connection string that was obtained when creating the namespace, and set **queueName** as the queue name that you used when creating the queue.

```

var connectionString = "";
var queueName = "samplequeue";

var client = QueueClient.CreateFromConnectionString(connectionString, queueName);

client.OnMessage(message =>
{
    Console.WriteLine(String.Format("Message body: {0}", message.GetBody<String>()));
    Console.WriteLine(String.Format("Message id: {0}", message.MessageId));
});

Console.ReadLine();

```

Here is what your Program.cs file should look like:

```

using System;
using Microsoft.ServiceBus.Messaging;

```

```

namespace GettingStartedWithQueues
{
    class Program
    {
        static void Main(string[] args)
        {
            var connectionString = "";
            var queueName = "samplequeue";

            var client = QueueClient.CreateFromConnectionString(connectionString, queueName);

            client.OnMessage(message =>
            {
                Console.WriteLine(String.Format("Message body: {0}", message.GetBody<String>()));
                Console.WriteLine(String.Format("Message id: {0}", message.MessageId));
            });

            Console.ReadLine();
        }
    }
}

```

4. Run the program, and check the portal. Notice that the **Queue Length** value should now be 0.

Congratulations! You have now created a queue, sent a message, and received a message.

Read [Get started with queues online](https://riptutorial.com/azureservicebus/topic/5882/get-started-with-queues): <https://riptutorial.com/azureservicebus/topic/5882/get-started-with-queues>

Credits

S. No	Chapters	Contributors
1	Getting started with azureservicebus	Community , Sean Feldman
2	Get started with queues	Jonathan Seaman