
















How to send and receive messages between .NET8 WebAPI and Azure ServiceBus

1. Create in Azure Portal a ServiceBus

We create a ServiceBus

The screenshot shows the Microsoft Azure portal interface. On the left sidebar, the 'Service Bus' resource is highlighted with a red box. Below it, the '+ Create' button is also highlighted with a red box. The main content area shows the 'Service Bus' resource page for the user 'particular (luiscoenriquez@hotmail.onmicrosoft.com)'. It includes a search bar, a 'Filter for any field...' input, and several filter buttons: 'Subscription equals all', 'Resource group equals all', and 'Location equals all'. Below the filters, it says 'Showing 0 to 0 of 0 records.' and displays a table with columns: Name, Status, Tier, Location, Resource group, and Subscription. The table is empty. Below the table, there is a message: 'No service bus namespaces to display' with a brief description of Azure Service Bus: 'Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics, used to decouple applications and services from each other.'

We input the ServiceBus data

 Create a resource Home Dashboard All services FAVORITES All resources Resource groups Quickstart Center App Services Function App SQL databases Azure Cosmos DB Virtual machines Load balancers Storage accounts Virtual networks Microsoft Entra ID Monitor Advisor Microsoft Defender for Cloud[Home](#) > [Service Bus](#) >

Create namespace

Service Bus

Basics[Advanced](#)[Networking](#)[Tags](#)[Review + create](#)

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Subscription 1



Resource group *

myRG

[Create new](#)

Instance Details

Enter required settings for this namespace.

Namespace name *

mythirdservicebus

.servicebus.windows.net

Location *

West Europe

Pricing tier *

Standard (~\$10 USD per 12.5M Operations per Month)

[Browse the available plans and their features](#)



Review + create

< Previous

Next: Advanced >

[Home](#) > [Service Bus](#) >

Create namespace



Service Bus



Validation succeeded.

[Basics](#)[Advanced](#)[Networking](#)[Tags](#)[Review + create](#)Service Bus Namespace
by Microsoft

Basics

Namespace name

mythirdservicebus

Subscription

Subscription 1

Resource group

myRG

Location

West Europe

Pricing tier

Standard

Networking

Connectivity method

Public access

Security

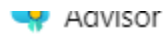
Minimum TLS version

1.2

Local Authentication

Enabled

[+ Create a resource](#)[Home](#)[Dashboard](#)[All services](#)[★ FAVORITES](#)[All resources](#)[Resource groups](#)[Quickstart Center](#)[App Services](#)[Function App](#)[SQL databases](#)[Azure Cosmos DB](#)[Virtual machines](#)[Load balancers](#)[Storage accounts](#)[Virtual networks](#)[Microsoft Entra ID](#)[Monitor](#)



ADVISOR

Microsoft Defender for
CloudCost Management +
Billing

Create

< Previous

Next >

We verify the created ServiceBus in the list

Microsoft Azure

Search resources, services, and docs (G+)

Home >

Service Bus

particular (luiscocoenriquezhotmail.onmicrosoft.com)

+ Create Manage view Refresh Export to CSV Open query Assign tags Delete

Filter for any field... Subscription equals all Resource group equals all Location equals all Add filter

Showing 1 to 1 of 1 records.

<input type="checkbox"/> Name ↑↓	Status ↑↓	Tier ↑↓	Location ↑↓	Resource group ↑↓	Subscription ↑↓	
<input type="checkbox"/> mythirdservicebus	Active	Standard	West Europe	myRG	Subscription 1	...

Microsoft Azure

Search resources, services, and docs (G+)

Home > Service Bus >

Service Bus

particular (luiscoenriquezhotmail.onmicrosoft.c...)

+ Create Manage view ...

Filter for any field...

Name ↑↓

- mythirdservicebus

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Shared access policies
- Geo-Recovery
- Migrate to premium
- Encryption
- Configuration
- Properties
- Locks

Entities

- Queues
- Topics

Monitoring

- Insights (Preview)

mythirdservicebus

Service Bus Namespace

Search

+ Queue + Topic Refresh Delete Give feedback

Essentials

JSON View

Resource group [\(move\)](#) [myRG](#)

Status: Active

Location: West Europe

Subscription [\(move\)](#) [Subscription 1](#)

Subscription ID: 9988cc6-c635-4ebd-b0ac-1be1dace0089

Tags [\(edit\)](#) [Add tags](#)

Created: Thursday, January 11, 2024

Updated: Thursday, January 11, 2024

Pricing tier: [Standard](#)

Host name: mythirdservicebus.servicebus.windows.net

Local Authentication: [Enabled](#)

Show data for the last: 1 hour 6 hours 12 hours 1 day 7 days 30 days

Requests

Line chart showing Requests over time. The y-axis ranges from 2 to 20. The chart shows several peaks, with the highest peak reaching approximately 18 requests.

Messages

Line chart showing Messages over time. The y-axis ranges from 0.5 to 2. The chart shows several peaks, with the highest peak reaching approximately 2 messages.

We create a new queue

Microsoft Azure

Search resources, services, and docs (G+/)

Home > mythirdservicebus | Overview > mythirdservicebus

mythirdservicebus | Queues ☆ ...

Service Bus Namespace

Search

+ Queue Refresh Give feedback

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Settings

Shared access policies
Geo-Recovery
Migrate to premium
Encryption
Configuration
Properties
Locks

Entities

Queues
Topics

Search to filter items by name...

Name	Status	Message count	Active messages	Dead-letter messages	Scheduled messages	Max size	Enabled
No results.							

Queues 0

The screenshot displays the Microsoft Azure portal interface. On the left, the navigation pane shows the 'mythirdservicebus' Service Bus Namespace selected under 'Queues'. The main content area shows the 'Create queue' dialog with the following configuration:

- Name:** myqueue1974
- Max queue size:** 1 GB
- Max delivery count:** 10
- Message time to live:** 14 Days, 0 Hours, 0 Minutes, 0 Seconds
- Lock duration:** 0 Days, 0 Hours, 1 Minutes, 0 Seconds
- Options (all unchecked):**
 - Enable auto-delete on idle queue
 - Enable duplicate detection
 - Enable dead lettering on message expiration
 - Enable partitioning
 - Enable sessions
 - Forward messages to queue/topic

The 'Create' button is visible at the bottom of the dialog.

This is the queue we created

The screenshot shows the Microsoft Azure portal interface. On the left is a navigation pane with options like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main area displays the 'Service Bus' page for the namespace 'mythirdservicebus'. A sidebar on the right lists various management options such as 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Settings', and 'Entities'. The 'Entities' section is expanded, showing 'Queues' and 'Topics'. The 'Queues' section contains a table with one entry: 'myqueue1974', which is 'Active' and has 0 messages.

Name	Status	Message count	Active messages	Dead-letter messages	Scheduled messages
myqueue1974	Active	0	0	0	0

We create a new topic

Microsoft Azure

Search resources, services, and docs (G+/)

Home > mythirdservicebus | Overview > mythirdservicebus

mythirdservicebus | Topics ☆ ...

Service Bus Namespace

Search

+ Topic Refresh Give feedback

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Settings

Shared access policies
Geo-Recovery
Migrate to premium
Encryption
Configuration
Properties
Locks

Entities

Queues
Topics

Search to filter items by name...

Topics 0

Name	Status	Scheduled messages	Max size	Subscription count	Enable partitioning
No results.					

Microsoft Azure

Search resources, services, and docs (G+)

Home > mythirdservicebus | Overview > mythirdservicebus

mythirdservicebus | Topics ☆ ...

Service Bus Namespace

Search

+ Topic Refresh Give feedback

Search to filter items by name...

Name	Status	Scheduled messages	Max size
No results.			

Create topic

Service Bus

Name * ⓘ

mytopiq ✓

Max topic size ⓘ

1 GB

Message time to live ⓘ

Days: 14 Hours: 0 Minutes: 0 Seconds: 0

☐ Enable auto-delete on idle topic ⓘ

☐ Enable duplicate detection ⓘ

☐ Enable partitioning ⓘ

☐ Support ordering ⓘ

Create Give feedback

This is the topic we created

The screenshot displays the Microsoft Azure portal interface. On the left, the 'FAVORITES' sidebar lists various services, including 'Service Bus'. The main content area is titled 'Service Bus' and shows the 'mythirdservicebus' namespace. A search bar is present, and a list of topics is displayed. The 'Topics' page for 'mythirdservicebus' is shown, featuring a search bar and a table of topics.

Name	Status	Scheduled messages	Max size	Subscription count	Enal
mytopic	Active	0	1024 MB	2	false

We have to click on the topic link to see the subscriptions

Microsoft Azure

Search resources, services, and docs (G+)

Home > mythirdservicebus | Overview > mythirdservicebus | Topics >

mytopic (mythirdservicebus/mytopic) Service Bus Topic

Search

+ Subscription Refresh Delete Give feedback

Overview

Access control (IAM)

Diagnose and solve problems

Service Bus Explorer

Settings

Shared access policies

Properties

Locks

Entities

Subscriptions

Automation

CLI / PS

Tasks (preview)

Export template

Help

Support + Troubleshooting

Essentials

Namespace : mythirdservicebus

Status : Active

Partitioning : Disabled

Duplicate detection : Disabled

Topic URL : <https://mythirdservicebus.servicebus.windows.net/mytopic>

Created : Thursday, January 11, 2024 at 14:04:24 GMT+1

Updated : Thursday, January 11, 2024 at 14:04:24 GMT+1

Support ordering : Disabled

Settings

Current size : 0.0 KB

Max size : 1 GB (change)

Message time to live : 14 DAYS (change)

Auto-delete : NEVER (change)

Free space : 100.0 %

Message count

Scheduled : 0 MESSAGES

Metrics

Show data for the last: 1 hour 6 hours 12 hours 1 day 7 days 30 days

Requests

Messages

Create a resource

Home

Dashboard

All services

FAVORITES

All resources

Resource groups

Quickstart Center

App Services

Function App

SQL databases

Azure Cosmos DB

Virtual machines

Load balancers

Storage accounts

Virtual networks

Microsoft Entra ID

Monitor

Advisor

Microsoft Defender for Cloud

[Home](#) > [mythirdservicebus | Overview](#) > [mythirdservicebus | Topics](#) > [mytopic \(mythirdservicebus/mytopic\)](#) >

Create subscription ...

Service Bus

Name * ⓘ

myfirstsubscription ✓

Max delivery count * ⓘ

10

Auto-delete after idle for ⓘ

Days

14

Hours

0

Minutes

0

Seconds

0

☐ Never auto-delete☐ Forward messages to queue/topic ⓘ

MESSAGE SESSIONS

Service bus sessions allow ordered handling of unbounded sequences of related messages. With sessions enabled a subscription can guarantee first-in-first-out delivery of messages. [Learn more.](#)

☐ Enable sessions

MESSAGE TIME TO LIVE AND DEAD-LETTERING

Message time to live (default) ⓘ

Days

14

Hours

0

Minutes

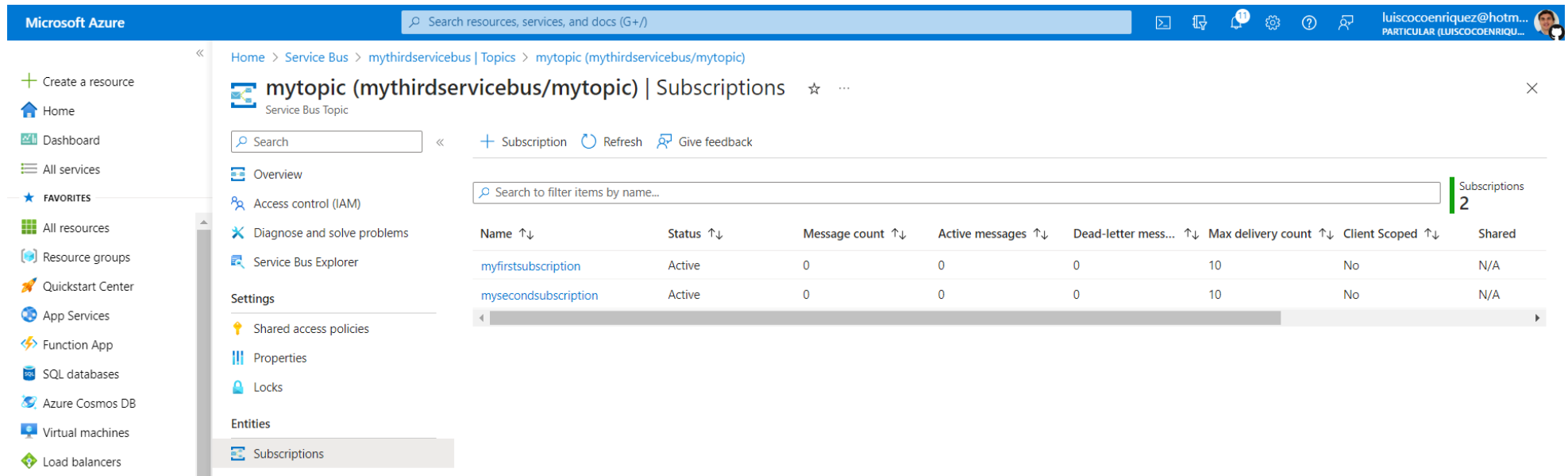
0

Seconds

0

☐ Enable dead lettering on message expiration

These are the two subscriptions



The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation options like 'Create a resource', 'Home', 'Dashboard', 'All services', and 'FAVORITES'. The main area displays the 'mytopic (mythirdservicebus/mytopic) | Subscriptions' page. It includes a search bar, a '+ Subscription' button, and a table listing two subscriptions: 'myfirstsubscription' and 'mysecondsubscription'. Both are in an 'Active' state with 0 messages and 0 active messages. The table also shows 'Dead-letter mess...', 'Max delivery count', 'Client Scoped', and 'Shared' columns.

Name ↑↓	Status ↑↓	Message count ↑↓	Active messages ↑↓	Dead-letter mess... ↑↓	Max delivery count ↑↓	Client Scoped ↑↓	Shared
myfirstsubscription	Active	0	0	0	10	No	N/A
mysecondsubscription	Active	0	0	0	10	No	N/A

2. Application configuration file (appsettings.json)

We include the appsettings.json file the: ServiceBus ConnectionString, QueueName and TopicName:

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "AzureServiceBus": {
```

```
"ConnectionString": "Endpoint=sb://mythirdservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;Sh
"QueueName": "myqueue1974",
"TopicName": "mytopic",
"FirstSubscriptionName": "myfirstsubscription",
"SecondSubscriptionName": "mysecondsubscription"
}
}
```

3. Configure the application Middleware (Program.cs)

We bind the ServiceBus configuration in the Program.cs file:

```
using AzureServiceBus.Model;

var builder = WebApplication.CreateBuilder(args);

// Bind Azure Service Bus settings
var azureServiceBusConfig = new ServiceBusConfig();
builder.Configuration.GetSection("AzureServiceBus").Bind(azureServiceBusConfig);
builder.Services.AddSingleton(azureServiceBusConfig);

// Add services to the container.
builder.Services.AddControllers();
builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();

var app = builder.Build();

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}
```



```
app.UseHttpsRedirection();

app.UseAuthorization();

app.MapControllers();

app.Run();
```

4. ServiceBus configuration Model (ServiceBusConfig.cs)

```
namespace AzureServiceBus.Model
{
    public class ServiceBusConfig
    {
        public string ConnectionString { get; set; }
        public string QueueName { get; set; }
        public string TopicName { get; set; }
        public string FirstSubscriptionName { get; set; }
        public string SecondSubscriptionName { get; set; }
    }
}
```

5. Create the application controller (ServiceBusWebApiControllers.cs)

```
using Microsoft.AspNetCore.Mvc;
using Azure.Messaging.ServiceBus;
using System.Text.Json;
using AzureServiceBus.Model;

namespace AzureServiceBus.Controllers
{
```

```
[ApiController]
[Route("[controller]")]
public class MessageController : ControllerBase
{
    private readonly ServiceBusConfig _config;

    public MessageController(ServiceBusConfig config)
    {
        _config = config;
    }

    [HttpPost("PostMessageToQueue")]
    public async Task<IActionResult> PostMessageToQueue([FromBody] object jsonData)
    {
        await using var client = new ServiceBusClient(_config.ConnectionString);
        ServiceBusSender sender = client.CreateSender(_config.QueueName);

        try
        {
            string messageBody = JsonSerializer.Serialize(jsonData);
            ServiceBusMessage message = new ServiceBusMessage(messageBody);
            await sender.SendMessageAsync(message);
            return Ok("Message sent to Azure Service Bus");
        }
        catch (Exception ex)
        {
            return StatusCode(500, $"Internal server error: {ex}");
        }
        finally
        {
            await sender.DisposeAsync();
        }
    }

    [HttpPost("PostMessageToTopic")]
    public async Task<IActionResult> PostMessageToTopic([FromBody] object jsonData)
    {

```

```
await using var client = new ServiceBusClient(_config.ConnectionString);
ServiceBusSender sender = client.CreateSender(_config.TopicName);

try
{
    string messageBody = JsonSerializer.Serialize(jsonData);
    ServiceBusMessage message = new ServiceBusMessage(messageBody);
    await sender.SendMessageAsync(message);
    return Ok("Message sent to Azure Service Bus Topic");
}
catch (Exception ex)
{
    return StatusCode(500, $"Internal server error: {ex}");
}
finally
{
    await sender.DisposeAsync();
}
}

[HttpGet("ReceiveMessageFromFirstSubscription")]
public async Task<IActionResult> ReceiveMessageFromFirstSubscription()
{
    await using var client = new ServiceBusClient(_config.ConnectionString);
    ServiceBusReceiver receiver = client.CreateReceiver(_config.TopicName, _config.FirstSubscriptionName);

    try
    {
        ServiceBusReceivedMessage receivedMessage = await receiver.ReceiveMessageAsync(TimeSpan.FromSeconds(10));
        if (receivedMessage != null)
        {
            string messageBody = receivedMessage.Body.ToString();
            await receiver.CompleteMessageAsync(receivedMessage);
            return Ok($"Received message: {messageBody}");
        }
        return NotFound("No message available in the subscription at this time.");
    }
}
```

```
        catch (Exception ex)
        {
            return StatusCode(500, $"Internal server error: {ex}");
        }
        finally
        {
            await receiver.DisposeAsync();
        }
    }

    [HttpGet("ReceiveMessageFromSecondSubscription")]
    public async Task<IActionResult> ReceiveMessageFromSecondSubscription()
    {
        await using var client = new ServiceBusClient(_config.ConnectionString);
        ServiceBusReceiver receiver = client.CreateReceiver(_config.TopicName, _config.SecondSubscriptionName);

        try
        {
            ServiceBusReceivedMessage receivedMessage = await receiver.ReceiveMessageAsync(TimeSpan.FromSeconds(10));
            if (receivedMessage != null)
            {
                string messageBody = receivedMessage.Body.ToString();
                await receiver.CompleteMessageAsync(receivedMessage);
                return Ok($"Received message: {messageBody}");
            }
            return NotFound("No message available in the subscription at this time.");
        }
        catch (Exception ex)
        {
            return StatusCode(500, $"Internal server error: {ex}");
        }
        finally
        {
            await receiver.DisposeAsync();
        }
    }
}
```

```
[HttpGet("ReceiveMessageFromQueue")]
public async Task<IActionResult> ReceiveMessageFromQueue()
{
    await using var client = new ServiceBusClient(_config.ConnectionString);
    ServiceBusReceiver receiver = client.CreateReceiver(_config.QueueName);

    try
    {
        ServiceBusReceivedMessage receivedMessage = await receiver.ReceiveMessageAsync(TimeSpan.FromSeconds(10));
        if (receivedMessage != null)
        {
            string messageBody = receivedMessage.Body.ToString();
            await receiver.CompleteMessageAsync(receivedMessage);
            return Ok($"Received message from queue: {messageBody}");
        }
        return NotFound("No message available in the queue at this time.");
    }
    catch (Exception ex)
    {
        return StatusCode(500, $"Internal server error: {ex}");
    }
    finally
    {
        await receiver.DisposeAsync();
    }
}
```

6. Verify the application

Swagger
Supported by SMARTBEAR

Select a definition **AzureServiceBus v1**

AzureServiceBus 1.0 OAS3

<https://localhost:7042/swagger/v1/swagger.json>

Message

POST	/Message/PostMessageToQueue	✓
POST	/Message/PostMessageToTopic	✓
GET	/Message/ReceiveMessageFromFirstSubscription	✓
GET	/Message/ReceiveMessageFromSecondSubscription	✓
GET	/Message/ReceiveMessageFromQueue	✓