

Azure Functionsで  
Service Bus メッセージの  
セッションIDを取得する

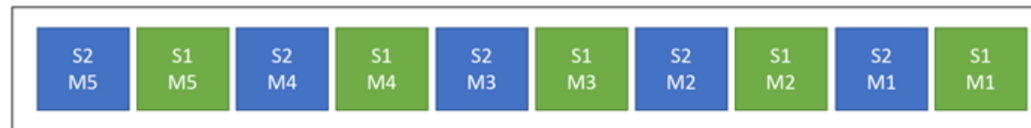
■ Service Busには「セッション」機能がある。詳しくは以下の公式サイトを参照。

<https://learn.microsoft.com/ja-jp/azure/service-bus-messaging/message-sessions>

■ 以下のサイトの説明がわかりやすい。

<https://jaliyaudagedara.blogspot.com/2021/04/azure-service-bus-handling-fifo-using.html>

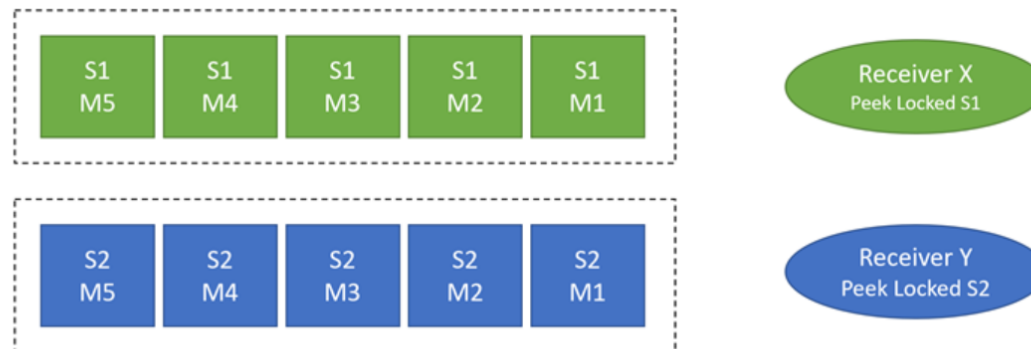
So basically what it does is, adding messages to the target queue in the following order. Note: I am setting the **SessionId** in each message.



Queue

Now if we have just one instance of a receiver/consumer, we should be ok. But what if we have multiple instances of receivers processing these messages. There is no way to guarantee the FIFO here. That's where the use of sessions comes into the picture.

When Sessions are enabled for a queue, behind the scene it will create virtual queues. And the receivers can either choose a particular session they are interested in and Peek Lock that entire Session or pick whatever the next session which isn't already locked by any other receivers. So that particular receiver will only be processing messages that belongs to the session they locked in. And no other receivers would see those locked in session messages.



Virtual Queues

## ■セッションを使用するにはStandardレベル以上のService Bus名前空間を作る必要がある

### ① 重要

- Service Bus の Basic レベルはセッションをサポートしていません。Standard レベルと Premium レベルはセッションをサポートしています。これらのレベルの違いについては、「[Service Bus の価格](#)」を参照してください。
- キューまたはサブスクリプションの作成後に、メッセージ セッションを有効または無効にすることはできません。それができるのはキューまたはサブスクリプションの作成時のみです。

<https://learn.microsoft.com/ja-jp/azure/service-bus-messaging/enable-message-sessions>

## ■ Standard または Premium レベルの、Service Bus名前空間を作る

Microsoft Azure

リソース、サービス、ドキュメントの検索 (G+/)

developer11@contoso1...  
CONTOSO11 (CONTOSO11993...

ホーム > Service Bus >

名前空間の作成

Service Bus

基本

詳細

ネットワーク

タグ

確認および作成

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション \*

Azure Pass - スポンサー プラン

リソース グループ \*

test1

新規作成

インスタンスの詳細

この名前空間に必要な設定を入力します。

名前空間の名前 \*

standard2983742

.servicebus.windows.net

場所 \*

Japan East

価格レベル \*

Standard (~\$10 USD (12.5M 操作あたりの月額))

使用可能なプランとその機能の参照

確認および作成

< 前へ

次へ: 詳細 >

価格レベルの選択

使用可能なプランとその機能の参照

★ お勧め | すべて表示

Basic	Standard
共有された容量	共有された容量
最大メッセージ サイズ 256 KB	最大メッセージ サイズ 256 KB
キュー	キュー
可変価格設定	トピック
	メッセージング操作 1300 万操作/月
	可変価格設定
0.05 米国ドル/百万/月 (見積もり)	10.00 米国ドル/1300万/月 (見積もり)

Premium

専用の容量
最大メッセージ サイズ 100 MB
キュー
トピック
メッセージング操作 Free
固定価格設定
ゾーン冗長 サポートあり
Recommended For Production Workloads

選択

## ■セッションを有効にしたキューを作成

ホーム > standard2983742

 **standard2983742** | キュー ...  
Service Bus 名前空間

+ キュー

最新の情報に更新

フィードバック

概要

アクティビティ ログ

アクセス制御 (IAM)

タグ

問題の診断と解決

設定

共有アクセス ポリシー

geo リカバリー

Premium へ移行

暗号化

構成

プロパティ

ロック

エンティティ

**キュー**

トピック

監視

分析情報 (プレビュー)

警告

名前	状態	最大
結果なし。		

キューの作成

Service Bus

名前 \* ⓘ  
queue1 ✓

最大キュー サイズ  
1 GB

最大配信数 \* ⓘ  
10

メッセージの Time to Live ⓘ  
日 時間 分 秒  
14 0 0 0

ロック期間 ⓘ  
日 時間 分 秒  
0 0 0 30

☐ アイドル キューの自動削除を有効にする ⓘ

☐ 重複データ検出を有効にする ⓘ

☐ メッセージの期限切れによる配信不能を有効にする ⓘ

☐ パーティション分割を有効にする ⓘ

☒ セッションを有効にする ⓘ

作成

## ■ Visual Studio で、プロジェクトを作成

追加情報

Azure Functions C# Azure クラウド

Functions worker ⓘ  
[.NET 6.0 (長期的なサポート)]

Function ⓘ  
[Service Bus Queue trigger]

☒ ランタイム ストレージ アカウントに Azurite を使用する (AzureWebJobsStorage) ⓘ  
☐ Docker を有効にする(E) ⓘ


Connection string setting name ⓘ  
[ ]

Queue name ⓘ  
[queue2]

戻る(B) 作成(C)

## ■Service Busの「プライマリ接続文字列」をコピーしておく

ホーム > Service Bus > sb2342345

 **sb2342345** | 共有アクセス ポリシー ☆ ...


Service Bus 名前空間

検索


<< + 追加

検索して項目をフィルター処理...

ポリシー

 RootManageSharedAccessKey

設定

 共有アクセス ポリシー

geo リカバリー

暗号化

構成

プロパティ

ロック

エンティティ

SAS ポリシー: RootManageSharedAccessKey

保存 破棄 削除 プライマリ キーの再生成 セカンダリ キーの再生成

☒ 管理

☒ 送信

☒ リッスン

主キー

yXYRjBB8/pm10aluRjKW7sil3DdkTGW7CiILGNWJo+U=

セカンダリ キー

7PjbdXsN6dmHwUVRhCSH152VpEqumV+YwmHUFgut70s=

プライマリ接続文字列

Endpoint=sb://sb2342345.servicebus.windows.net/;SharedAccessKeyName=RootMana...

セカンダリ接続文字列

Endpoint=sb://sb2342345.servicebus.windows.net/;SharedAccessKeyName=RootMana...

SAS ポリシー ARM ID

/subscriptions/d005e678-2df0-4d0f-9c98-a3802c5b8d98/resourcegroups/rg4/provider...

- 「local.settings.json」を開き、  
「AzureWebJobsServiceBus」の値として、コピーしておいた「プライマリ接続文字列」を設定

※なお、このファイルは Git のバージョン管理の対象外とするように .gitignore で指定されている。



```
1 {
2   "IsEncrypted": false,
3   "Values": {
4     "AzureWebJobsStorage": "UseDevelopmentStorage=true",
5     "FUNCTIONS_WORKER_RUNTIME": "dotnet",
6     "AzureWebJobsServiceBus": "Endpoint=sb://sb2342345.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey"
7   }
8 }
```



## ■関数のコードを書き換える

Function1.cs 4 local.settings.json

FunctionApp10 FunctionApp10.Function1 Run

```
1 using System;
2 using Microsoft.Azure.ServiceBus;
3 using Microsoft.Azure.WebJobs;
4 using Microsoft.Azure.WebJobs.Host;
5 using Microsoft.Extensions.Logging;
6
7 namespace FunctionApp10
8 {
9     0 個の参照
10    public class Function1
11    {
12        [FunctionName("Function1")]
13        0 個の参照
14        public void Run([ServiceBusTrigger("queue1", IsSessionsEnabled=true)] Message myQueueItem, ILogger log)
15        {
16            log.LogInformation($"C# ServiceBus queue trigger function processed message: {myQueueItem}");
17            log.LogInformation($"SessionId: {myQueueItem.SessionId}");
18        }
19    }
20 }
```

usingディレクティブを追加

トリガーで、  
キューの名前と、  
IsSessionsEnabledを設定

第1引数の型を  
string型から  
Message型に変更

第1引数のオブジェクト  
のSessionIdプロパ  
ティを参照できる

## ■ Service Busキューにメッセージを送信（適当なセッションIDを設定）

ホーム > standard2983742 | キュー > queue1 (standard2983742/queue1)

### queue1 (standard2983742/queue1) | Service Bus エクスプローラー

Service Bus キュー

検索

- 概要
- アクセス制御 (IAM)
- 問題の診断と解決
- Service Bus エクスプローラー

設定

- 共有アクセス ポリシー
- プロパティ
- ロック

オートメーション

- タスク (プレビュー)
- テンプレートのエクスポート

ヘルプ

- 新しいサポート リクエスト

クイック表示モード ▾ ▶ メッセージの送信

キュー (0) 配信不能 (0)

最初からクイック表示 → 次のメッセージを

0 件中 0 件のメッセージを表示

シーケンス番号      メッセージ ID

ここにメッセージを表示

### メッセージの送信

メッセージ本文

1 hello

コンテンツの種類

text/plain ▾

✓ ブローカーのプロパティ

関連付け ID

メッセージ ID

返信先

ラベルまたは件名

セッション ID

1234

☐ Time to Live を設定する

☐ スケジュールされたエンキュー時間を設定する

☐ 送信の繰り返し

送信      キャンセル

## ■ 動作確認

```
C:\Users\hiryamada\AppData \ + v

Azure Functions Core Tools
Core Tools Version:      4.0.4915 Commit hash: N/A (64-bit)
Function Runtime Version: 4.14.0.19631

[2022-12-20T11:30:48.778Z] Found C:\Users\hiryamada\source\repos\FunctionApp10\FunctionApp10.csproj. Using for user secrets file configuration.

Functions:

    Function1: serviceBusTrigger

For detailed output, run func with --verbose flag.
[2022-12-20T11:30:56.718Z] Host lock lease acquired by instance ID '00000000000000000000000003BC4ABDC'.
[2022-12-20T11:31:02.995Z] Executing 'Function1' (Reason='(null)', Id=3806f90b-5148-49b4-aeb6-98c421e4a21a)
[2022-12-20T11:31:03.000Z] Trigger Details: MessageId: 6df3d94ceb6b4827a9129cddbccc15b19, SequenceNumber: 1, DeliveryCount: 1, EnqueuedTimeUtc: 2022-12-20T11:31:03.0380000Z, LockedUntilUtc: 9999-12-31T14:59:59.9999999Z, SessionId: 1234
[2022-12-20T11:31:03.028Z] C# ServiceBus queue trigger function processed message: {MessageId:6df3d94ceb6b4827a9129cddbccc15b19}
[2022-12-20T11:31:03.030Z] SessionId: 1234
[2022-12-20T11:31:03.059Z] Executed 'Function1' (Succeeded, Id=3806f90b-5148-49b4-aeb6-98c421e4a21a, Duration=171ms)
```

セッションのstate情報を  
取得・更新する

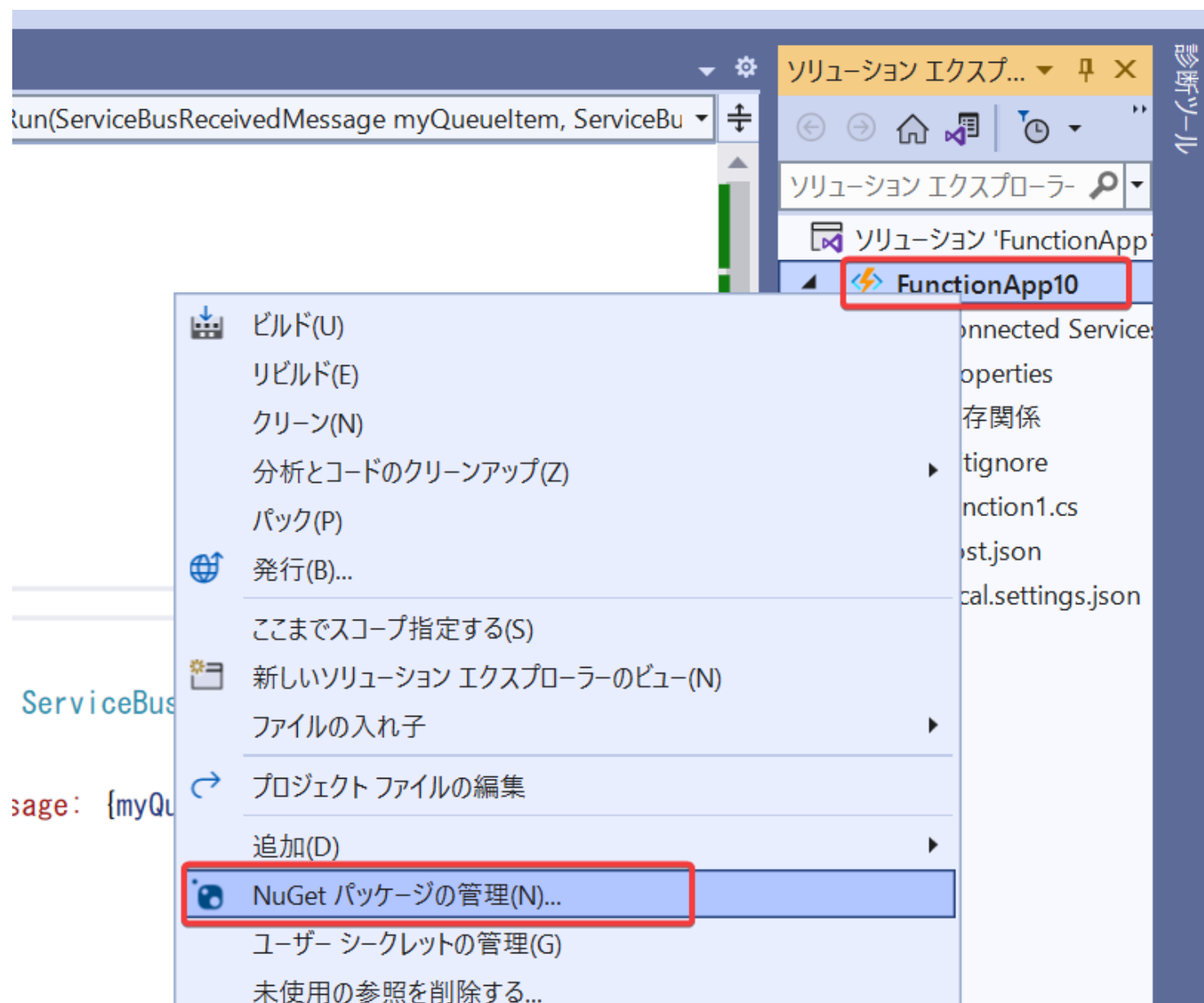
■以下のサイトにサンプルがある。

[https://github.com/Azure/azure-sdk-for-net/blob/main/sdk/servicebus/Azure.Messaging.ServiceBus/samples/Sample06\\_Transactions.md#setting-session-state-within-a-transaction](https://github.com/Azure/azure-sdk-for-net/blob/main/sdk/servicebus/Azure.Messaging.ServiceBus/samples/Sample06_Transactions.md#setting-session-state-within-a-transaction)

■ただしこのサンプルは、  
「Microsoft.Azure.ServiceBus」ではなく、より新しい  
「Azure.Messaging.ServiceBus」を使用したコードとなっている。

■Azure Functionプロジェクトで使用されている「Service Bus拡張機能」のバージョンを最新化することで、「Azure.Messaging.ServiceBus」が使用できる。

## ■プロジェクトのパッケージの管理を開く




■ デフォルトでは拡張機能の4系統が使われている。これを5系統を使用するように更新する。

Function1.cs\* 新機能 FunctionApp10\* NuGet: FunctionApp10 local.settings.json

参照 インストール済み 更新プログラム 2 NuGet パッケージ


検索 (Ctrl+L) ☐ プレリリースを含める


Top-level packages (2)




Microsoft.Azure.WebJobs.Extensions.ServiceBus

作成者: Microsoft

4.3.1 

5.8.1 




Microsoft.NET.Sdk.Functions

作成者: Microsoft

4.1.1

4.1.3

Transitive packages (160)




Microsoft.AspNet.WebApi.Client

作成者: Microsoft

5.2.4

This package adds support for formatting and content negotiation to System.Net.Http.



Microsoft.AspNetCore.Authentication.Abstractions

作成者: Microsoft

2.1.0

■参考: Azure Functionsのドキュメントでも、「拡張機能 v5.x」が最新となっている。

拡張機能 v5.x

Functions 2.x 以降

Functions 1.x

メタデータを含むメッセージを送信する場合は、`ServiceBusMessage` 型を使います。パラメーターは `return` 型の属性として定義されます。複数のメッセージを書き込むには、`ICollector<T>` または `IAsyncCollector<T>` を使います。 `Add` メソッドを呼び出すときに、メッセージが作成されます。

関数が終了したときにパラメーター値が `null` の場合、Functions はメッセージを作成しません。

[ServiceBusAccountAttribute](#) 属性を使用して、使用する Service Bus アカウントを指定することもできます。コンストラクターは、Service Bus 接続文字列を含むアプリ設定の名前を受け取ります。属性は、パラメーター、メソッド、またはクラスレベルで適用できます。次の例では、クラスレベルとメソッドレベルを示します。

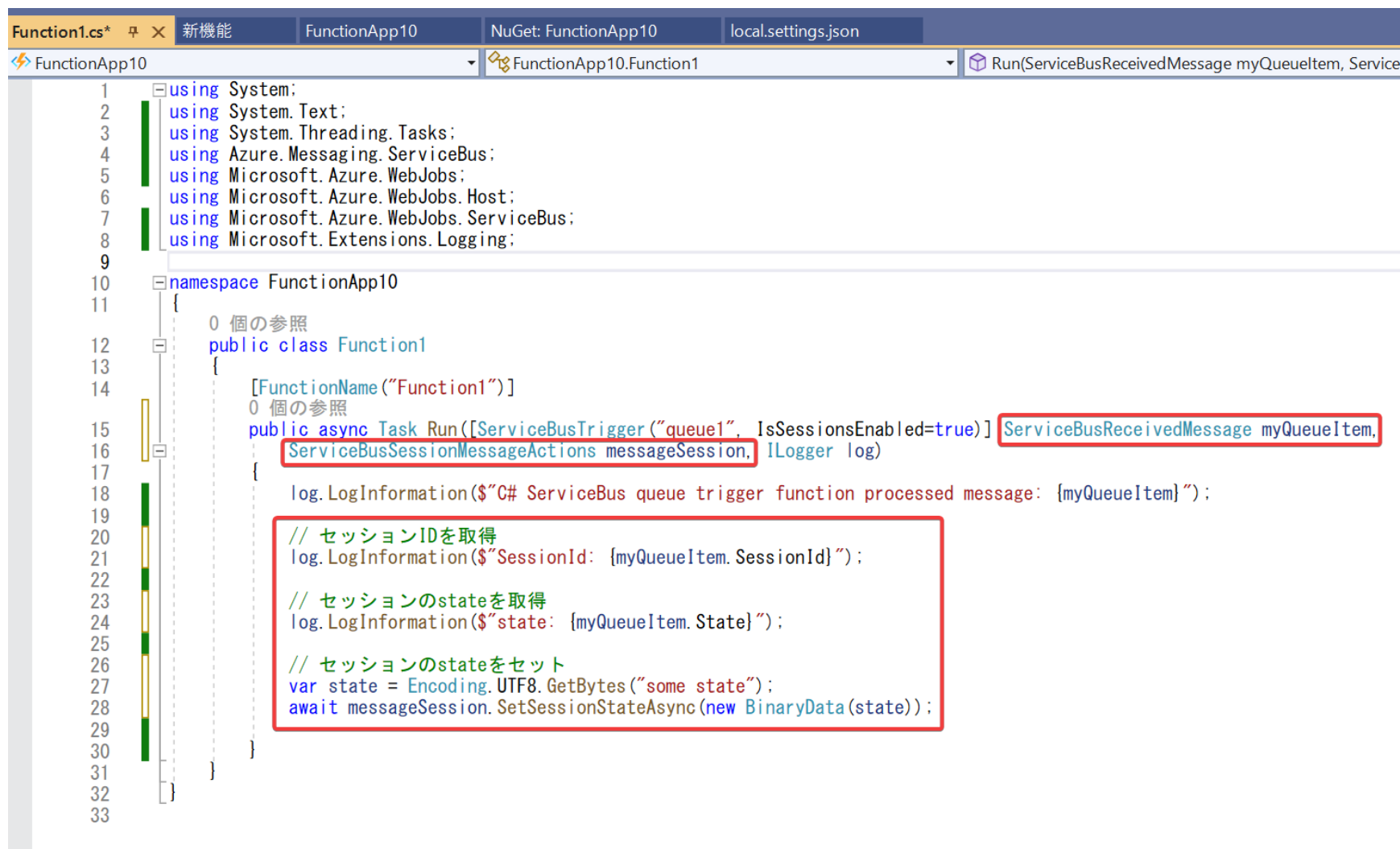
C#

コピー

```
[ServiceBusAccount("ClassLevelServiceBusAppSetting")]
public static class AzureFunctions
{
    [ServiceBusAccount("MethodLevelServiceBusAppSetting")]
    [FunctionName("ServiceBusQueueTriggerCSharp")]
    public static void Run(
        [ServiceBusTrigger("myqueue", AccessRights.Manage)]
        string myQueueItem, ILogger log)
    {
        ...
    }
}
```



■ 「Azure.Messaging.ServiceBus」に含まれるクラスを使用してセッションのstateを取得・更新することができる。またセッションIDも取得できる。



```
1 using System;
2 using System.Text;
3 using System.Threading.Tasks;
4 using Azure.Messaging.ServiceBus;
5 using Microsoft.Azure.WebJobs;
6 using Microsoft.Azure.WebJobs.Host;
7 using Microsoft.Azure.WebJobs.ServiceBus;
8 using Microsoft.Extensions.Logging;
9
10 namespace FunctionApp10
11 {
12     0 個の参照
13     public class Function1
14     {
15         [FunctionName("Function1")]
16         0 個の参照
17         public async Task Run([ServiceBusTrigger("queue1", IsSessionsEnabled=true)] ServiceBusReceivedMessage myQueueItem,
18             ServiceBusSessionMessageActions messageSession, ILogger log)
19         {
20             log.LogInformation($"C# ServiceBus queue trigger function processed message: {myQueueItem}");
21
22             // セッションIDを取得
23             log.LogInformation($"SessionId: {myQueueItem.SessionId}");
24
25             // セッションのstateを取得
26             log.LogInformation($"state: {myQueueItem.State}");
27
28             // セッションのstateをセット
29             var state = Encoding.UTF8.GetBytes("some state");
30             await messageSession.SetSessionStateAsync(new BinaryData(state));
31         }
32     }
33 }
```

[https://github.com/Azure/azure-sdk-for-net/blob/main/sdk/servicebus/Azure.Messaging.ServiceBus/samples/Sample06\\_Transactions.md#setting-session-state-within-a-transaction](https://github.com/Azure/azure-sdk-for-net/blob/main/sdk/servicebus/Azure.Messaging.ServiceBus/samples/Sample06_Transactions.md#setting-session-state-within-a-transaction)