

Snabbguide

Azure Functions och administrera IoT-Devices

Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone a repository

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)

Create a new project

Recent project templates

- Blank App (Universal Windows) C#
- Windows Forms App (.NET Framework) C#
- Azure Functions C#
- ASP.NET Core Web Application C#
- Console App (.NET Core) C#
- Blazor App C#
- ASP.NET Web Application (.NET Framework) C#
- Analysis Services Multidimensional and Data Mining Project
- Integration Services Project
- Windows Forms App (.NET Framework)

Visual Basic

azure functions

C# All platforms All project types

Azure Functions
A template to create an Azure Function project.
Azure Cloud C#

Azure Cloud Service (classic)
A project for creating a scalable service that runs on Microsoft Azure.
Azure Cloud C#

Azure Resource Group
This template creates an Azure Resource Group deployment project. The deployment project will contain artifacts needed to provision Azure resources using Azure Resource Manager that will create an environment for your application.
Azure Cloud C#

Azure WebJob (.NET Framework)
A project template for creating WebJobs which allow you to run programs in your Azure Web Apps.
Azure Cloud C#

Storm Azure SQL Writer Sample

Back Next

Configure your new project

Azure Functions

Azure

Cloud

C#

Project name

AzureFunctions

Location



Solution name 

AzureFunctions

Place solution and project in the same directory

Back

Create

Create a new Azure Functions application

Azure Functions v3 (.NET Core)



Empty

Creates an Azure Function project with no triggers. Function triggers can be added during development.



Blob trigger

A C# function that will be run whenever a blob is added to a specified container.



Cosmos DB Trigger

A C# function that will be run whenever documents change in a document collection.



Event Grid trigger

A C# function that will be run whenever an event grid receives a new event



Event Hub trigger

A C# function that will be run whenever an event hub receives a new event



Http trigger

A C# function that will be run whenever it receives an HTTP request

Storage account (AzureWebJobsStorage)

Storage emulator

None

Storage emulator

Browse...

[Get started with Azure Functions](#)

Back

Create

Create

Azure Storage

Select existing or create new

Subscription

Free Trial

Name	Resource group	Location
storageacountrgrou87c3	RGROUP	West Europe

 Create a storage account

Add Cancel Refresh Create

Create a new Azure Functions application

Azure Functions v3 (.NET Core)



Empty

Creates an Azure Function project with no triggers. Function triggers can be added during development.



Blob trigger

A C# function that will be run whenever a blob is added to a specified container.



Cosmos DB Trigger

A C# function that will be run whenever documents change in a document collection.



Event Grid trigger

A C# function that will be run whenever an event grid receives a new event



Event Hub trigger

A C# function that will be run whenever an event hub receives a new event



Http trigger

A C# function that will be run whenever it receives an HTTP request

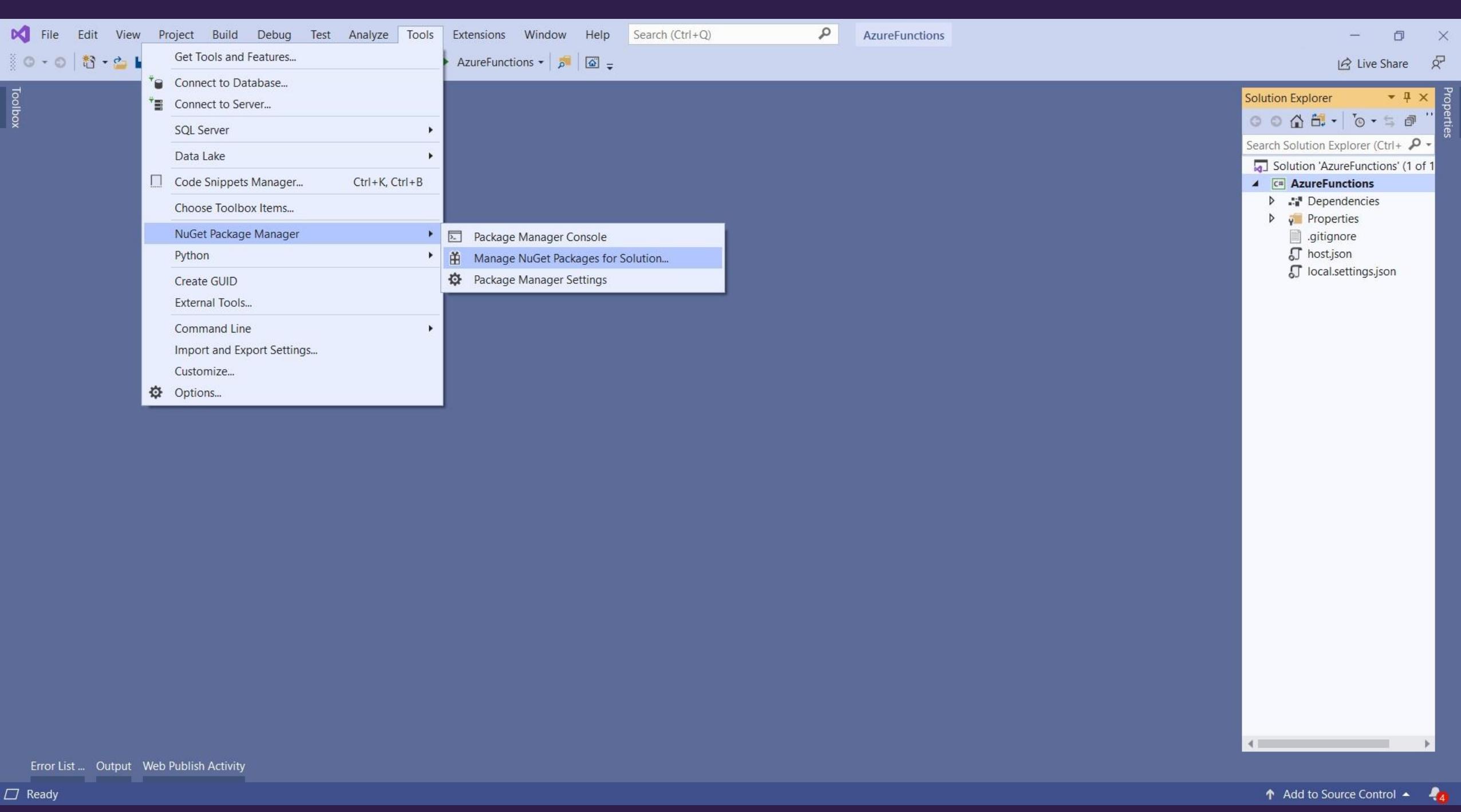
[Get started with Azure Functions](#)

Storage account (AzureWebJobsStorage)

storageaccountrgrou87c3

Back

Create



Screenshot of the Microsoft Visual Studio IDE showing the NuGet Manager interface for an Azure Functions project.

The top navigation bar includes File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q), and AzureFunctions.

The toolbar includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.

The Solution Explorer shows a single solution named "AzureFunctions" containing a project named "AzureFunctions".

The NuGet Manager window is open, showing the "Browse" tab selected (circled in red).

The search bar contains "Microsoft.azure.devices".

The results list includes:

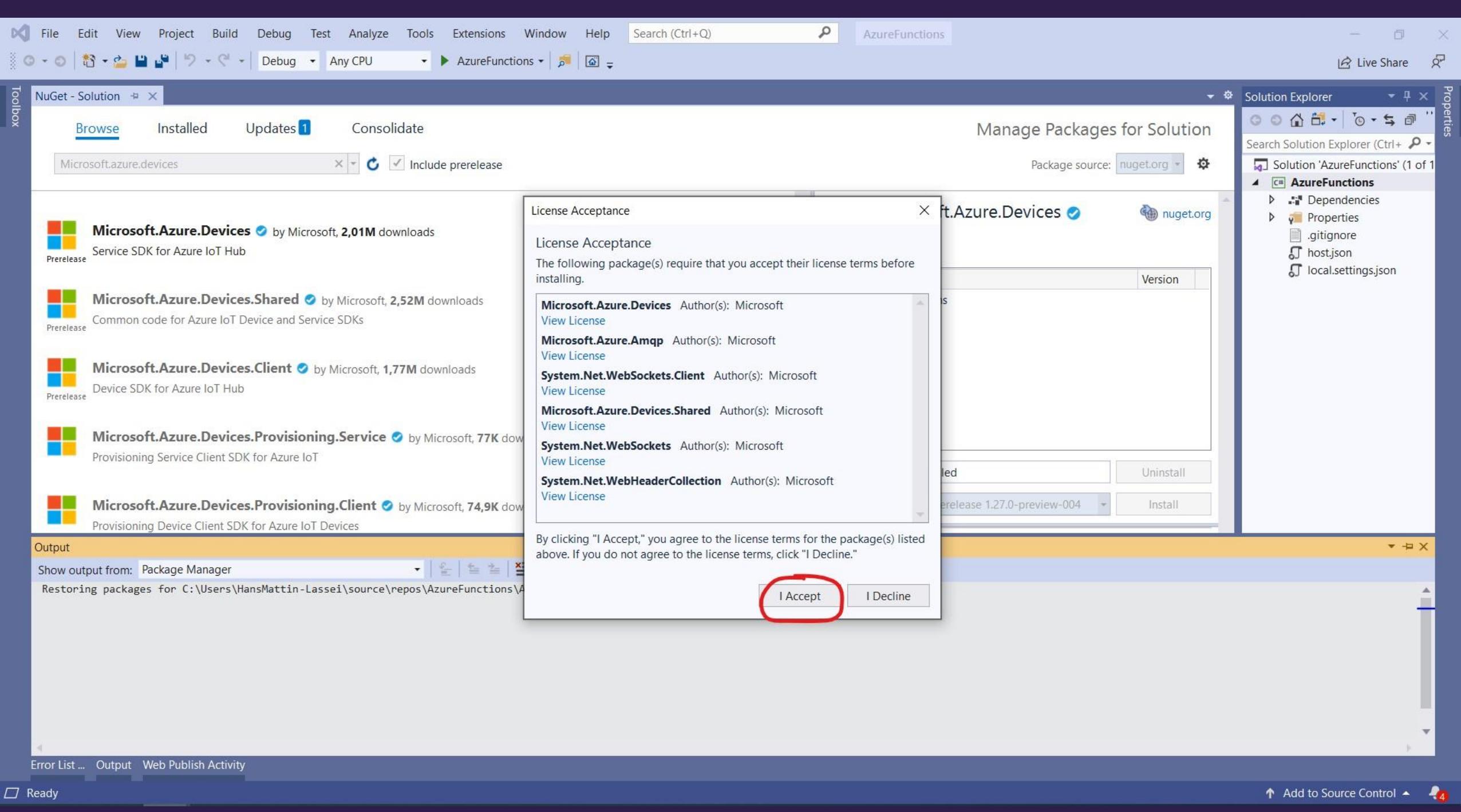
- Microsoft.Azure.Devices** by Microsoft, 2,01M downloads (Prerelease) - Version v1.27.0-preview-004
- Microsoft.Azure.Devices.Shared** by Microsoft, 2,52M downloads (Prerelease) - Version v1.25.0-preview-002
- Microsoft.Azure.Devices.Client** by Microsoft, 1,77M downloads (Prerelease) - Version v1.29.0-preview-004
- Microsoft.Azure.Devices.Provisioning.Service** by Microsoft, 77K downloads - Version v1.6.2
- Microsoft.Azure.Devices.Provisioning.Client** by Microsoft, 74,9K downloads - Version v1.5.2
- Microsoft.Azure.Devices.Client.PCL** by Microsoft, 37,3K downloads - Version v1.0.16
- Microsoft.Azure.Devices.Provisioning.Transport.Amqp** by Microsoft, 41,3K downloads - Version v1.2.2
- Microsoft.Azure.Devices.Provisioning.Transport.Mqtt** by Microsoft, 41K downloads - Version v1.2.2

The right pane displays the details for "Microsoft.Azure.Devices". It shows the package version as 1.27.0-preview-004, the source as nuget.org, and the fact that it is a Prerelease version. It also lists the projects where the package is installed: "Project" and "AzureFunctions".

The "Installed" status is "not installed". The "Version" dropdown is set to "Latest prerelease 1.27.0-preview-004". The "Install" button is highlighted with a red circle.

The "Options" section provides the package's description, version (1.27.0-preview-004), author (Microsoft), license (View License), publication date (Thursday, November 21, 2019 (11/21/2019)), and project URL (<https://github.com/Azure/azure-iot-sdk-csharp>).

The bottom navigation bar includes Error List ..., Output, Web Publish Activity, and Add to Source Control.



Home > IoT Hub >

unique-iothub-name | Shared access policies



IoT Hub

Search (Ctrl+ /)

<<

Add

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Shared access policies

Identity

Pricing and scale

Networking

Certificates

Built-in endpoints

Failover

Properties

Locks

Export template

Explorers



IoT Hub uses permissions to grant access to each IoT hub endpoint. Permissions limit the access to an IoT hub based on functionality.



Search to filter items...

Policy

iothubowner

service

device

registryRead

registryReadWrite

Permissions

registry write, service connect, device

service connect

device connect

registry read

registry write

iothubowner

unique-iothub-name

Save Discard Regenerate keys Delete

Access policy name

iothubowner

Permissions

Registry read

Registry write

Service connect

Device connect

Shared access keys

Primary key



Copy to clipboard

Secondary key



Connection string—primary key

HostName=unique-iothub-name.azure-devices.net...



Connection string—secondary key

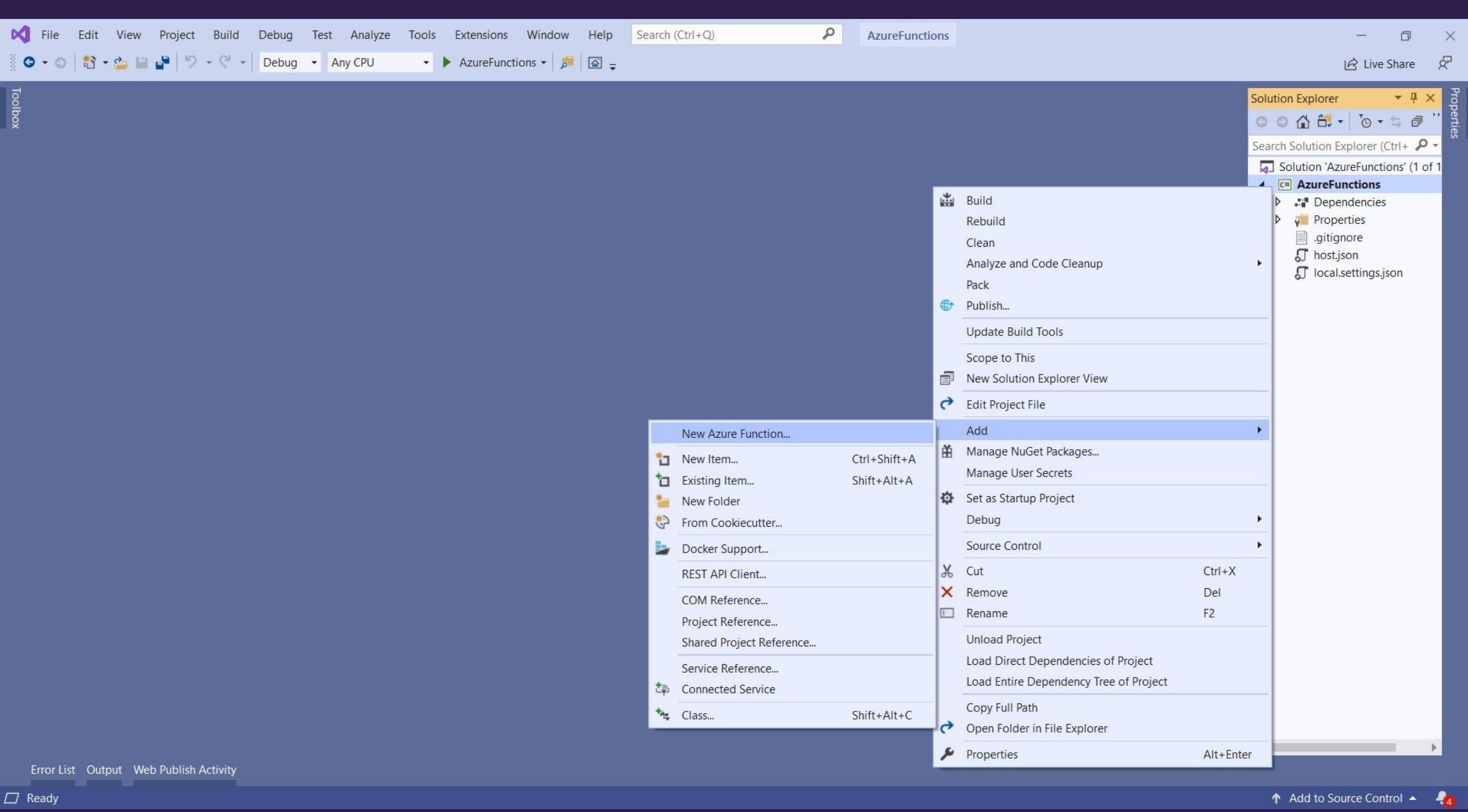


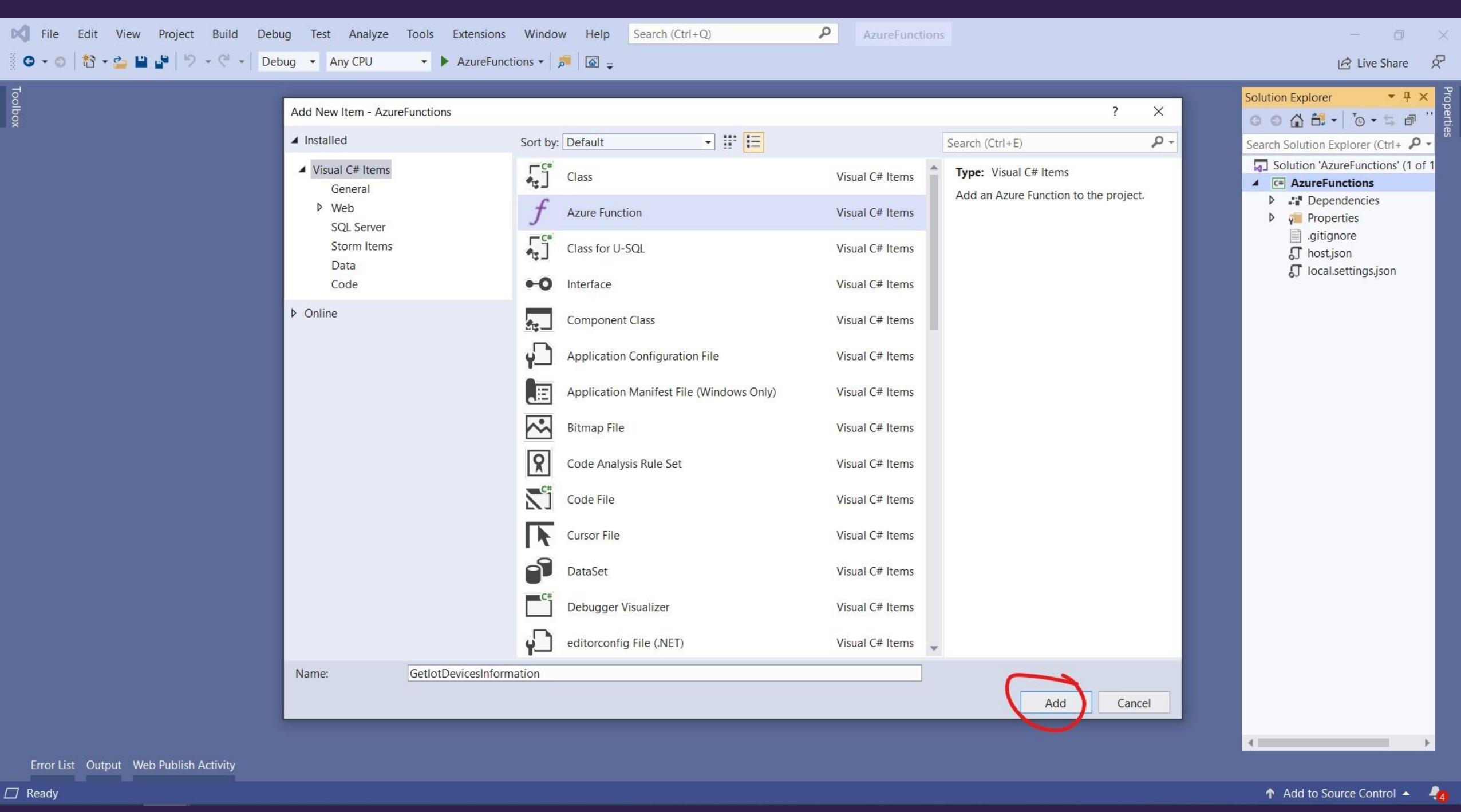
The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

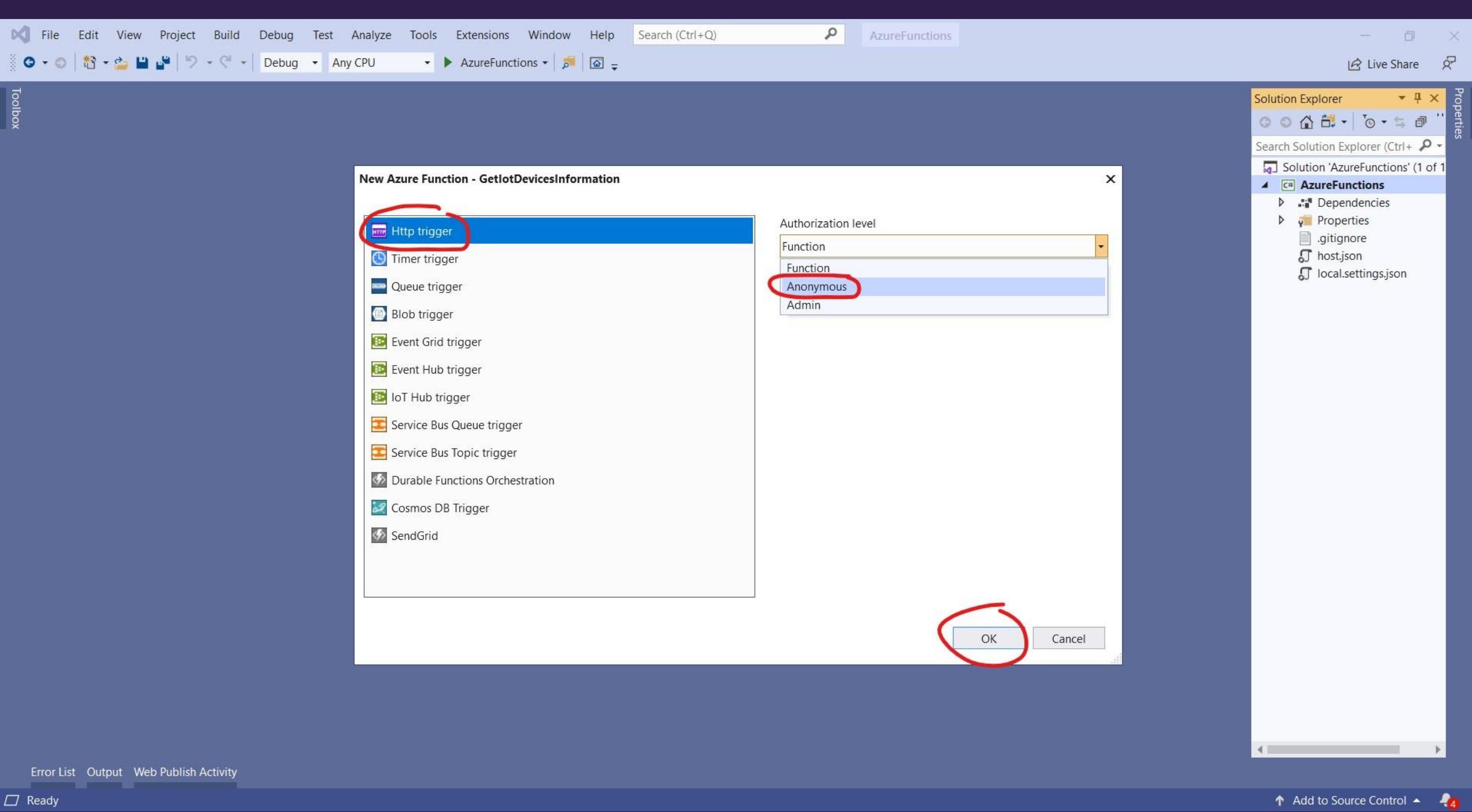
- File Menu:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q) with a magnifying glass icon.
- Toolbox:** Toolbox tab.
- Solution Explorer:** Shows the solution 'AzureFunctions' (1 of 1) with the following structure:
 - AzureFunctions
 - Dependencies
 - Properties
 - .gitignore
 - host.json
 - local.settings.json
- Properties:** Properties tab.
- Code Editor:** The file 'local.settings.json' is open, showing the following JSON content:

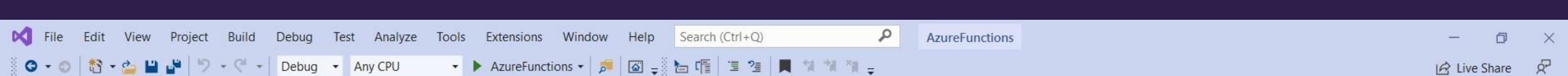
```
1 {  
2     "IsEncrypted": false,  
3     "Values": {  
4         "AzureWebJobsStorage": "DefaultEndpointsProtocol=https;AccountName=storageaccountrgrou87c3;AccountKey=m2gbU/BXprbtQMWAgwNI5FN76SM0Ykd5xQWv+39",  
5         "FUNCTIONS_WORKER_RUNTIME": "dotnet",  
6         "IoTHubConnection": "HostName=unique-iothub-name.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=jw1TMZGtqU/Fjvw4iAqSx56xr4"  
7     }  
8 }
```

A red arrow points from the 'local.settings.json' entry in the Solution Explorer down to the file in the code editor.
- Status Bar:** 110 %, No issues found, Ln: 6 Ch: 169 SPC CRLF.
- Bottom Navigation:** Error List, Output, Web Publish Activity, Ready, Add to Source Control, and a notification icon with a red '4'.









GetIoTDevicesInformation.cs

C# AzureFunctions

```
4  using Microsoft.AspNetCore.Mvc;
5  using Microsoft.Azure.WebJobs;
6  using Microsoft.Azure.WebJobs.Extensions.Http;
7  using Microsoft.AspNetCore.Http;
8  using Microsoft.Extensions.Logging;
9  using Newtonsoft.Json;
10 
11 namespace AzureFunctions
12 {
13     public static class GetIoTDevicesInformation
14     {
15         [FunctionName("GetIoTDevicesInformation")]
16         public static async Task<IActionResult> Run(
17             [HttpTrigger(AuthorizationLevel.Anonymous, "get", "post", Route = null)] HttpRequest req,
18             ILogger log)
19         {
20             log.LogInformation("C# HTTP trigger function processed a request.");
21 
22             string name = req.Query["name"];
23 
24             string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
25             dynamic data = JsonConvert.DeserializeObject(requestBody);
26             name = name ?? data?.name;
27 
28             string responseMessage = string.IsNullOrEmpty(name)
29                 ? "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized"
30                 : $"Hello, {name}. This HTTP triggered function executed successfully.";
31 
32             return new OkObjectResult(responseMessage);
33         }
34     }
35 }
```

110% ✓ No issues found

Error List Output Web Publish Activity

Ready

Add to Source Control

4

Solution Explorer

Search Solution Explorer (Ctrl + F)

Solution 'AzureFunctions' (1 of 1)

✓ AzureFunctions

- Dependencies
- Properties
- .gitignore
- GetIoTDevicesInformation
- host.json
- local.settings.json

Properties

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AzureFunctions

GetIoTDevicesInformation.cs* ▾ X

C# AzureFunctions AzureFunctions.GetIoTDevicesInformation Run(HttpContext req, ILogger log)

```
10 11  using Microsoft.Azure.Devices;
12 13  using System.Linq;
14 15  namespace AzureFunctions
16 17  {
18 19      public static class GetIoTDevicesInformation
20 21      {
22 23          private static readonly RegistryManager registryManager =
24 25              RegistryManager.CreateFromConnectionString(Environment.GetEnvironmentVariable("IoTHubConnectionString"));
26
27          [FunctionName("GetIoTDevicesInformation")]
28          public static async Task<IActionResult> Run(
29              [HttpTrigger(AuthorizationLevel.Anonymous, "get", Route = null)] HttpRequest req,
30              ILogger log)
31          {
32
33              var result = registryManager.CreateQuery($"SELECT * FROM devices");
34              var data = await result.GetNextAsTwinAsync();
35
36              if (data.Any())
37              {
38                  var json = JsonConvert.SerializeObject(data);
39                  return new OkObjectResult(json);
40              }
41              else
42              {
43                  return new NotFoundResult();
44              }
45          }
46      }
47  }
```

Toolbox

Solution Explorer

Properties

Search Solution Explorer (Ctrl+F)

Solution 'AzureFunctions' (1 of 1)

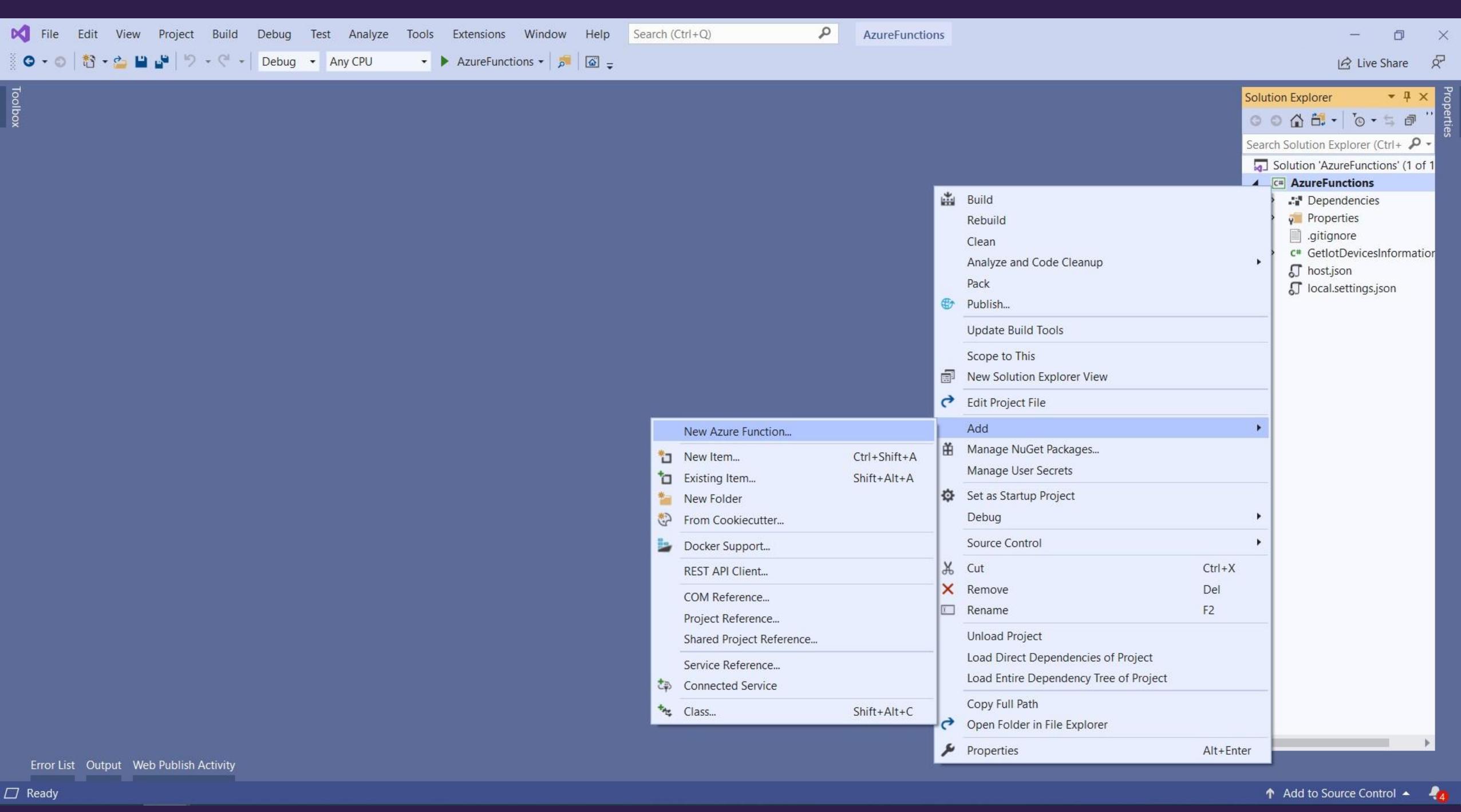
- Dependencies
- Properties
- .gitignore
- GetIoTDevicesInformation.cs
- host.json
- local.settings.json

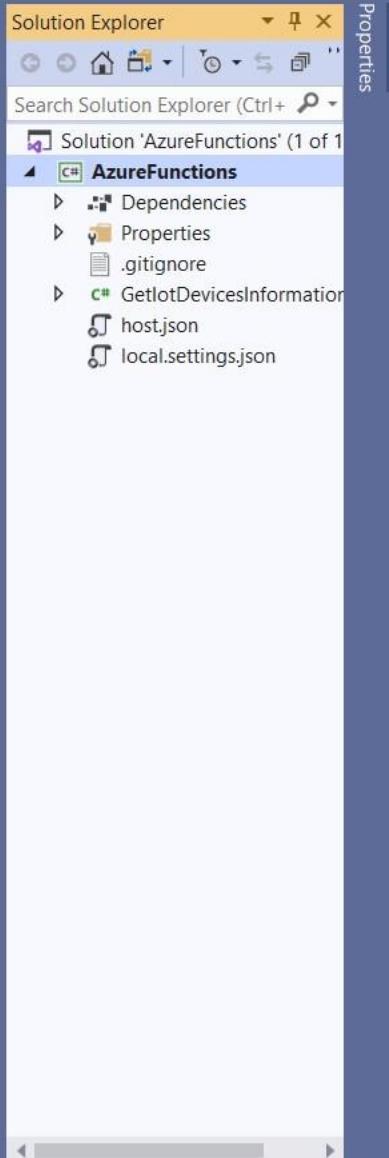
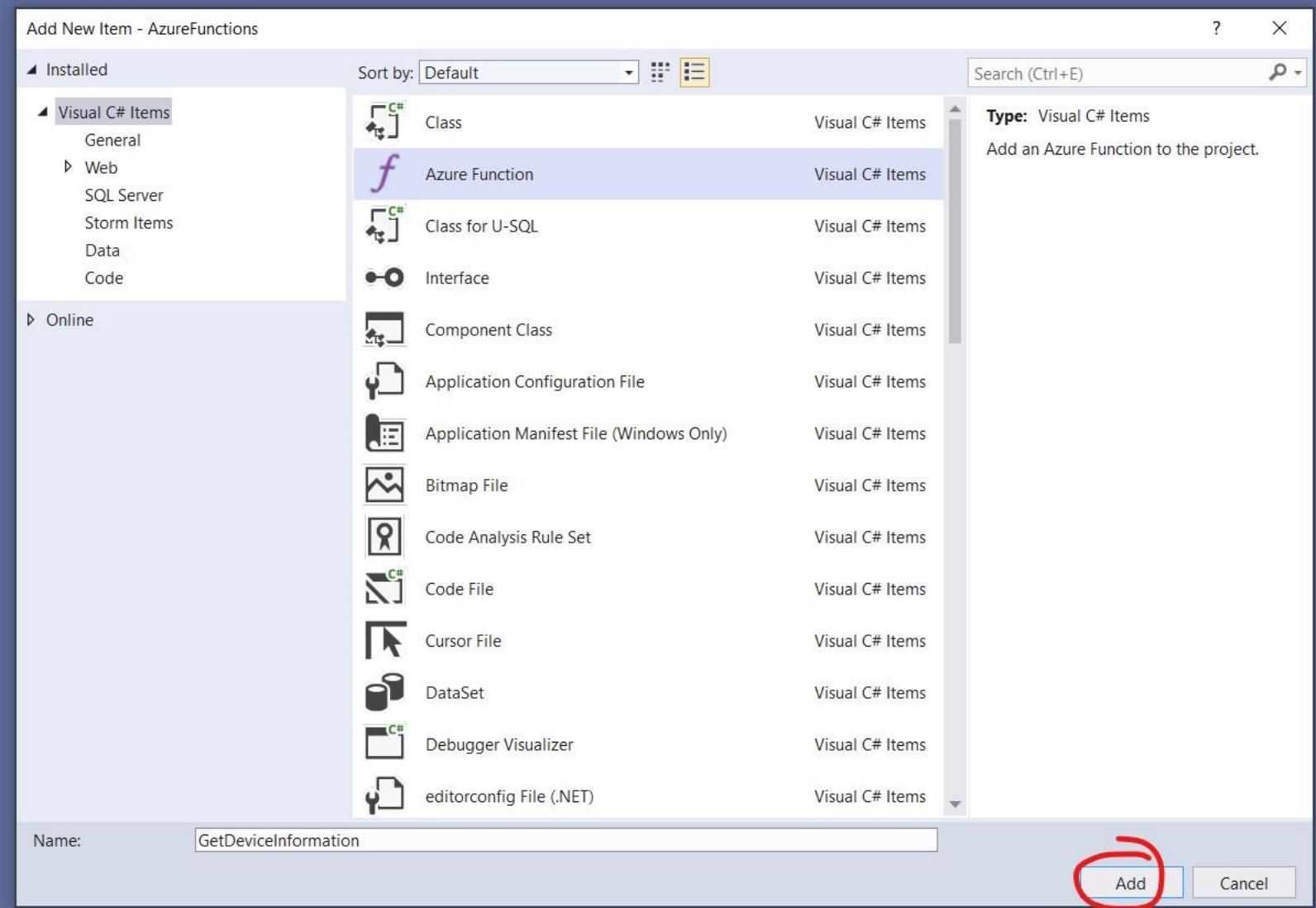
108 % No issues found Ln: 41 Ch: 10 SPC CRLF

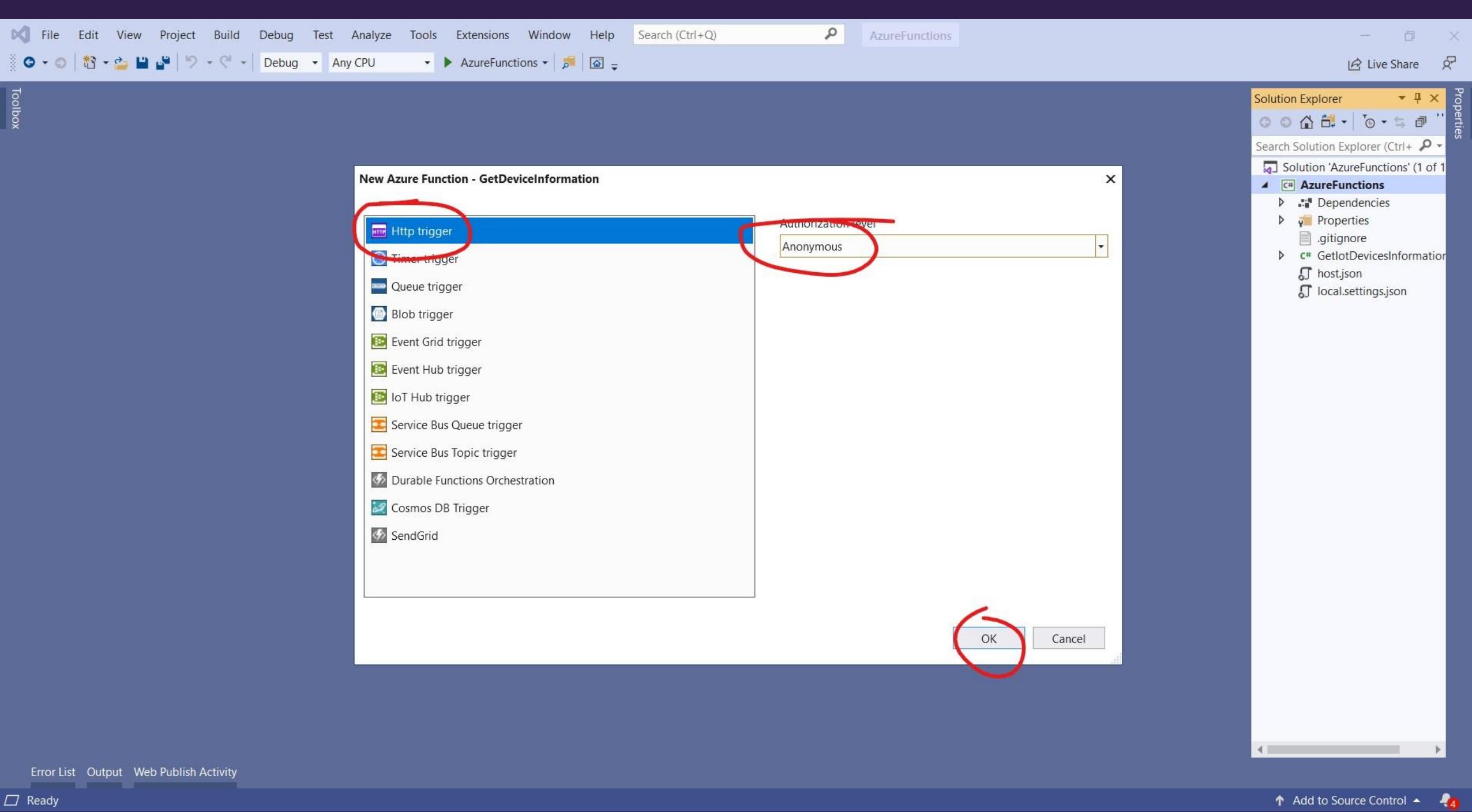
Error List Output Web Publish Activity

Add to Source Control

Ready







The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the solution 'AzureFunctions' with one item: 'AzureFunctions'.
- Properties:** Shows the properties for the 'AzureFunctions' item.
- Code Editor:** The file 'GetDeviceInformation.cs' contains the following C# code for an Azure Function:

```
4  using Microsoft.AspNetCore.Mvc;
5  using Microsoft.Azure.WebJobs;
6  using Microsoft.Azure.WebJobs.Extensions.Http;
7  using Microsoft.AspNetCore.Http;
8  using Microsoft.Extensions.Logging;
9  using Newtonsoft.Json;
10 
11 namespace AzureFunctions
12 {
13     public static class GetDeviceInformation
14     {
15         [FunctionName("GetDeviceInformation")]
16         public static async Task<IActionResult> Run(
17             [HttpTrigger(AuthorizationLevel.Anonymous, "get", "post", Route = null)] HttpRequest req,
18             ILogger log)
19         {
20             log.LogInformation("C# HTTP trigger function processed a request.");
21 
22             string name = req.Query["name"];
23 
24             string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
25             dynamic data = JsonConvert.DeserializeObject(requestBody);
26             name = name ?? data?.name;
27 
28             string responseMessage = string.IsNullOrEmpty(name)
29                 ? "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response"
30                 : $"Hello, {name}. This HTTP triggered function executed successfully.";
31 
32             return new OkObjectResult(responseMessage);
33         }
34     }
35 }
36 
```

A large red 'X' is drawn over the entire code block, indicating it is incorrect or incomplete.

Status Bar: 108 %, No issues found, Ln: 20, Ch: 1, SPC, CRLF.

Bottom Navigation: Error List, Output, Web Publish Activity, Ready, Add to Source Control.

File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) AzureFunctions

GetDeviceInformation.cs* | Debug Any CPU | AzureFunctions | Run(HttpRequest req, ILogger log)

Live Share

Toolbox

C# AzureFunctions

```
11 using Microsoft.Azure.Devices;
12 using System.Linq;
13
14 namespace AzureFunctions
15 {
16     public static class GetDeviceInformation
17     {
18
19         private static readonly RegistryManager registryManager =
20             RegistryManager.CreateFromConnectionString(Environment.GetEnvironmentVariable("IoTHubConnectionString"));
21
22         [FunctionName("GetDeviceInformation")]
23         public static async Task<IActionResult> Run(
24             [HttpTrigger(AuthorizationLevel.Anonymous, "get", Route = null)] HttpRequest req,
25             ILogger log)
26         {
27
28             string deviceid = req.Query["deviceid"];
29             var result = registryManager.CreateQuery($"SELECT * FROM devices WHERE deviceId = '{deviceid}'");
30             var data = await result.GetNextAsTwinAsync();
31
32             if (data.FirstOrDefault() != null)
33             {
34                 var json = JsonConvert.SerializeObject(data.FirstOrDefault());
35                 return new OkObjectResult(json);
36             }
37             else
38             {
39                 return new NotFoundResult();
40             }
41
42         }
43     }
}
```

Solution Explorer

Search Solution Explorer (Ctrl+F)

Solution 'AzureFunctions' (1 of 1)

- ✓ AzureFunctions
 - Dependencies
 - Properties
 - .gitignore
 - GetDeviceInformation.cs
 - GetlotDevicesInformation
 - host.json
 - local.settings.json

Properties

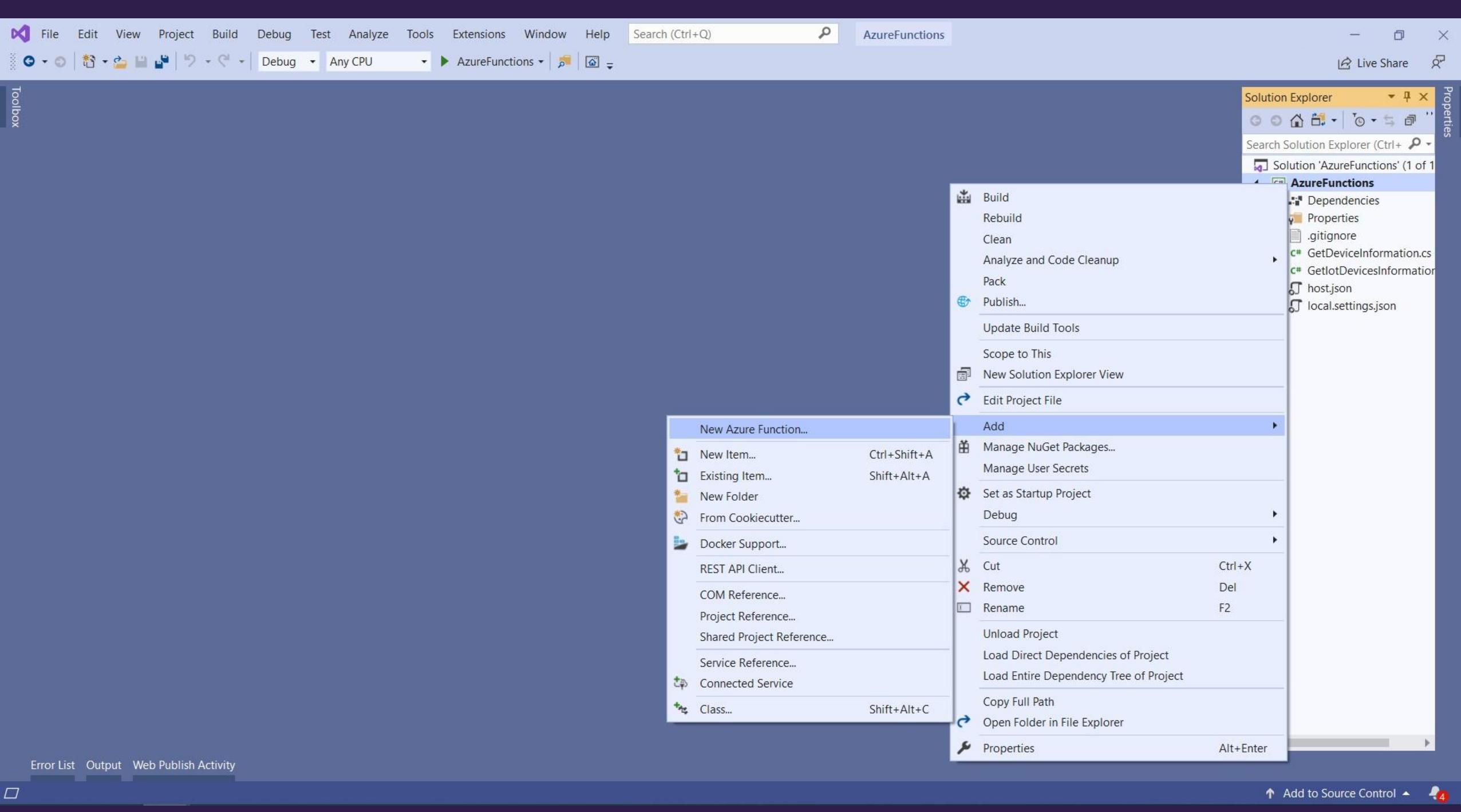
108 % No issues found

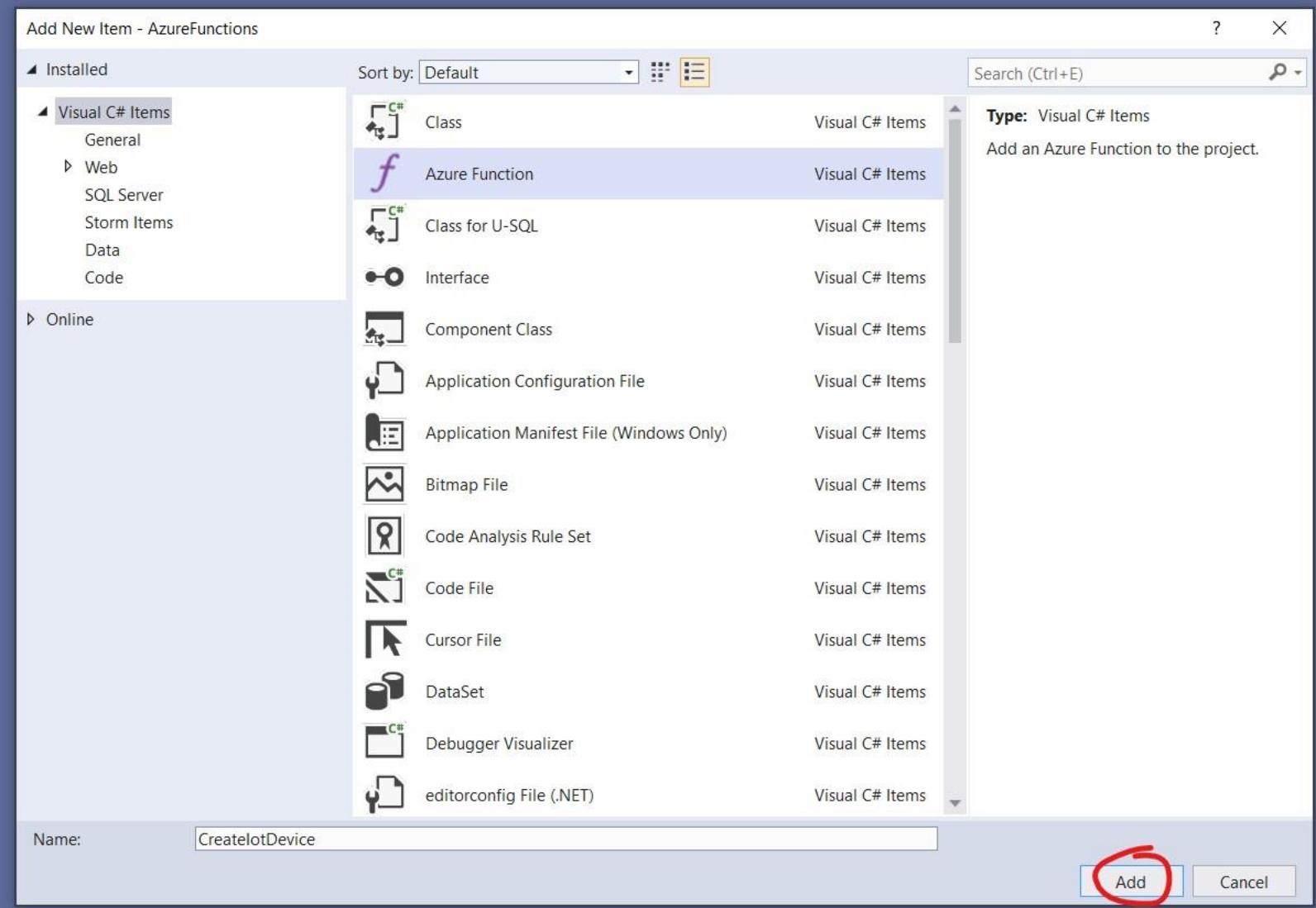
Error List Output Web Publish Activity

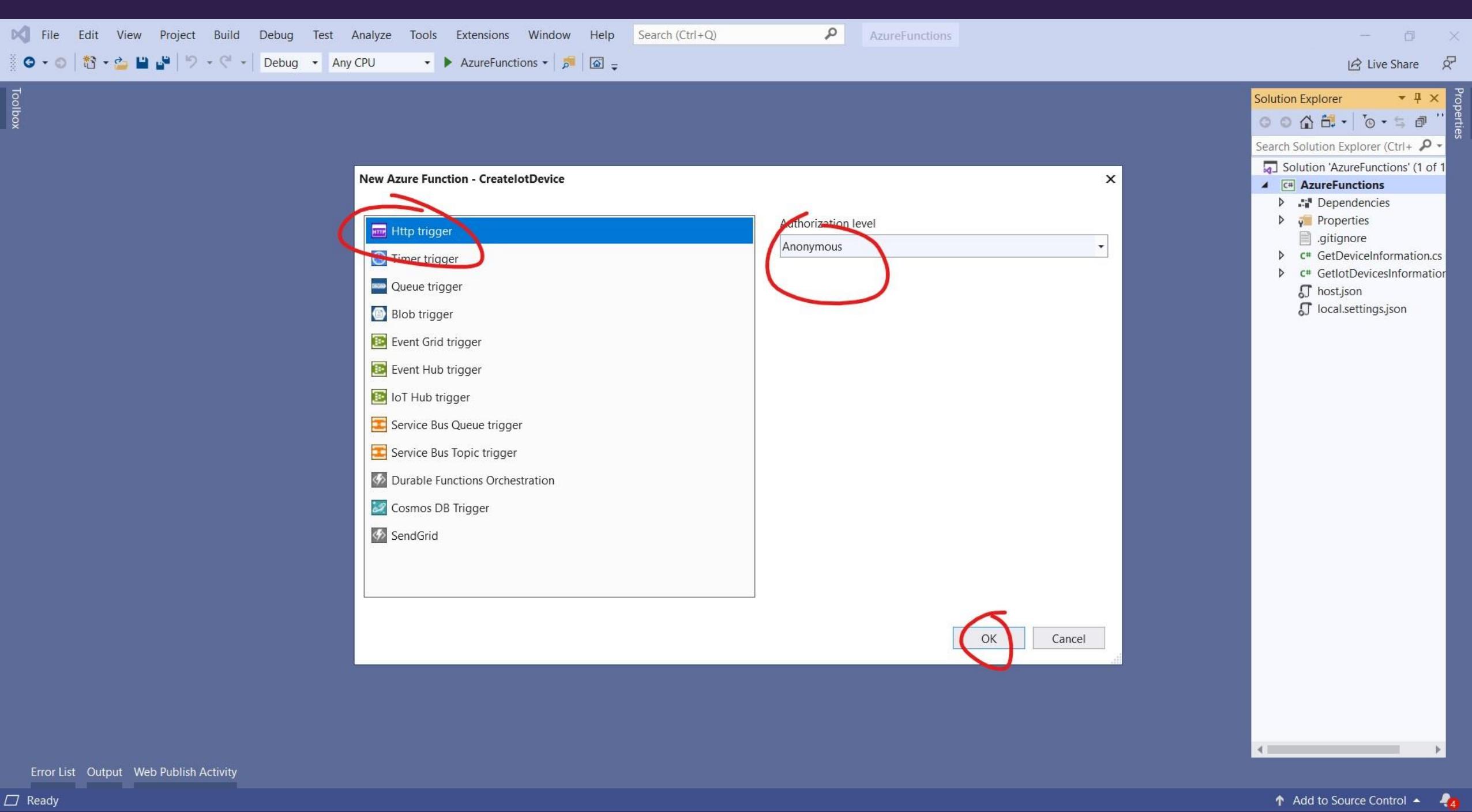
Ln: 43 Ch: 6 SPC CRLF

Add to Source Control

Ready







The screenshot shows the Microsoft Visual Studio IDE interface with the following details:

- File Menu:** File, Edit, View, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+Q).
- Solution Explorer:** Shows the solution 'AzureFunctions' with one item: 'AzureFunctions'.
- Properties:** Shows the properties for the 'AzureFunctions' project.
- Toolbox:** Standard development tools.
- Code Editor:** Displays the file 'CreateIotDevice.cs' with the following code:

```
4  using Microsoft.AspNetCore.Mvc;
5  using Microsoft.Azure.WebJobs;
6  using Microsoft.Azure.WebJobs.Extensions.Http;
7  using Microsoft.AspNetCore.Http;
8  using Microsoft.Extensions.Logging;
9  using Newtonsoft.Json;
10 
11 namespace AzureFunctions
12 {
13     public static class CreateIotDevice
14     {
15         [FunctionName("CreateIotDevice")]
16         public static async Task<IActionResult> Run(
17             [HttpTrigger(AuthorizationLevel.Anonymous, "get", "post", Route = null)] HttpRequest req,
18             ILogger log)
19         {
20             log.LogInformation("C# HTTP trigger function processed a request.");
21 
22             string name = req.Query["name"];
23 
24             string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
25             dynamic data = JsonConvert.DeserializeObject(requestBody);
26             name = name ?? data?.name;
27 
28             string responseMessage = string.IsNullOrEmpty(name)
29                 ? "This HTTP triggered function executed successfully. Pass a name in the query string or in the request body for a personalized response"
30                 : $"Hello, {name}. This HTTP triggered function executed successfully.";
31 
32             return new OkObjectResult(responseMessage);
33         }
34     }
35 }
36 
```

A large red 'X' is drawn across the entire code block, starting from the first line and ending at the last line. The code itself is functional and demonstrates how to handle an HTTP trigger in an Azure Function.

Status Bar: 108 %, No issues found, Ln: 20, Ch: 1, SPC, CRLF.

Bottom Navigation: Error List, Output, Web Publish Activity, Ready, Add to Source Control.

The screenshot shows the Microsoft Visual Studio IDE interface for an Azure Functions project named 'AzureFunctions'. The code editor displays the file 'CreateIotDevice.cs' containing C# code for an Azure Function.

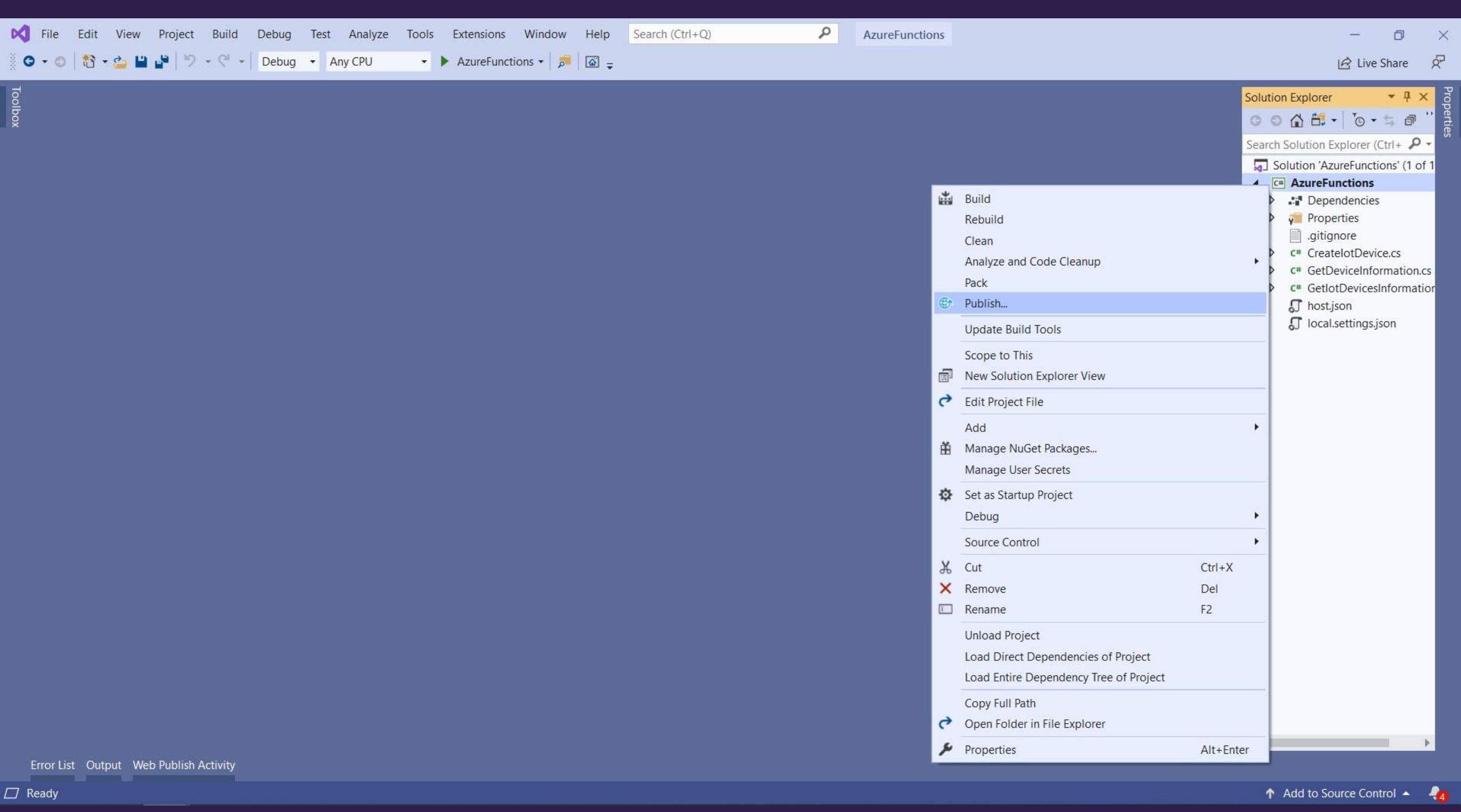
```
10  using Microsoft.Azure.Devices; ←
11
12  namespace AzureFunctions
13  {
14      public static class CreateIotDevice
15      {
16          private static readonly RegistryManager registryManager =
17              RegistryManager.CreateFromConnectionString(Environment.GetEnvironmentVariable("IoTHubConnectionString"));
18
19          [FunctionName("CreateIotDevice")]
20          public static async Task<IActionResult> Run(
21              [HttpTrigger(AuthorizationLevel.Anonymous, "post", Route = null)] HttpRequest req,
22              ILogger log)
23          {
24
25              string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
26              var data = JsonConvert.DeserializeObject<dynamic>(requestBody);
27
28              if (data != null)
29              {
30                  await registryManager.AddDeviceAsync(new Device(data.deviceId));
31                  return new OkResult();
32              }
33              else
34              {
35                  return new BadRequestResult();
36              }
37
38          }
39      }
40  }
```

The code implements an Azure Function named 'CreateIotDevice' that handles POST requests. It uses the `Microsoft.Azure.Devices` namespace and the `RegistryManager` class to interact with an IoT Hub. The function reads the request body as JSON and adds a new device to the registry if the parsed data is not null.

Annotations in the code:

- A red arrow points to the `using Microsoft.Azure.Devices;` statement at line 10.
- A red arrow points to the `RegistryManager.CreateFromConnectionString` call at line 17.
- A large red oval highlights the main processing logic starting from line 25, specifically the `string requestBody` assignment and the subsequent `if (data != null)` conditional block.

The Solution Explorer on the right shows the project structure, including files like `Dependencies`, `Properties`, `.gitignore`, and several C# files for different functions.



Live Share

AzureFunctions

Toolbox

Solution Explorer

Properties

Connected Services

Publish

Publish

Where are you publishing today?

Target

Azure
Publish your application to the Microsoft cloud

Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images

Folder
Publish your application to a local folder or file share

Import Profile
Import your publish settings to deploy your app

Back Next Finish Cancel

The screenshot shows the 'Publish' dialog in Visual Studio. The 'Azure' option is circled in red, and the 'Next' button at the bottom is also circled in red. The 'Docker Container Registry', 'Folder', and 'Import Profile' options are also visible.

Live Share

AzureFunctions

Toolbox

Solution Explorer

Properties

Connected Services

Publish

Publish

Which Azure service would you like to use to host your application?

Target

Specific target

Azure Function App (Windows)

Published your application code to a serverless compute that scales dynamically and runs code on-demand...

Azure Function App (Linux)

Published your application code to a serverless compute that scales dynamically and runs code on-demand...

Azure Function App Container

Published your application as a Docker image to Azure Container Registry and run it on Azure Function...

Azure Container Registry

Published your application as a Docker image to Azure Container Registry

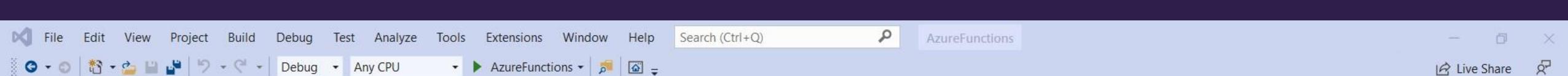
Back Next Finish Cancel

Search Solution Explorer (Ctrl+ Shift+F)

Solution 'AzureFunctions' (1 of 1)

✓ AzureFunctions

- Dependencies
- Properties
- .gitignore
- CreateIoTDevice.cs
- GetDeviceInformation.cs
- GetIoTDevicesInformation.cs
- host.json
- local.settings.json



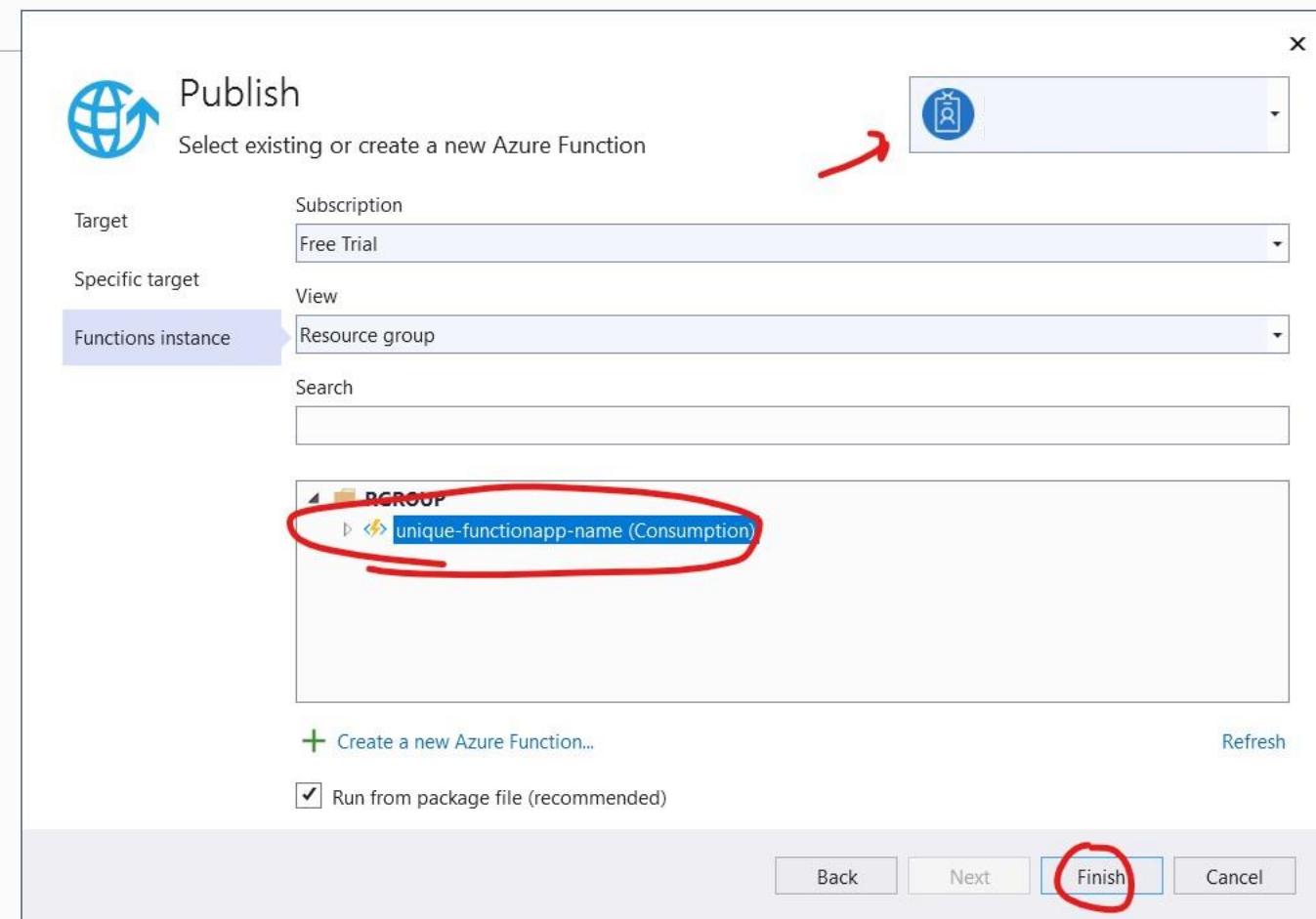
Toolbox

AzureFunctions

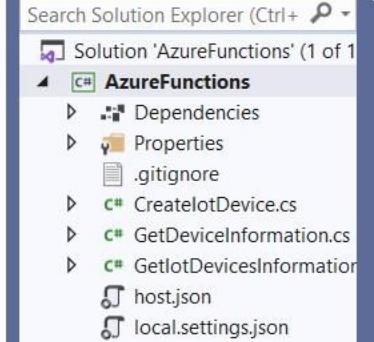
Connected Services

Publish

Publish



Solution Explorer



Properties

Error List Output Web Publish Activity

Ready

Add to Source Control

4

Screenshot of the Azure Functions Publish dialog in Visual Studio.

The dialog is titled "Configure dependency" and instructs the user to "Select a service dependency to add to your application." A red arrow points to the "Select" button (a dropdown menu icon).

The "Subscription" dropdown shows "Free Trial".

A table lists a single dependency:

Name	Resource group	Location
storageaccountrgrou87c3	RGROUP	West Europe

The "storageaccountrgrou87c3" entry is circled in red.

At the bottom, there are "Back", "Next", "Finish", and "Cancel" buttons. The "Next" button is circled in red.

In the Solution Explorer pane, the "AzureFunctions" project is selected. A red circle highlights the "Configure" link under the project summary, which has a yellow warning icon.

The Properties pane is visible on the right.

Toolbox, Error List, Output, Web Publish Activity, and Ready status bars are at the bottom.

AzureFunctions

Toolbox

AzureFunctions

Configure dependency

Provide connection string name and specify how to save it

Connection string name: AzureWebJobsStorage

Connection string value:

Tip: avoid pasting application secrets directly into your code.

Save connection string value in: [Learn more](#)

Azure App Settings

Azure Key Vault

None

Back Next **Finish** Cancel

Solution Explorer

Search Solution Explorer (Ctrl+ F)

AzureFunctions

- Dependencies
- Properties
- .gitignore
- CreatelotDevice.cs
- GetDeviceInformation.cs
- GetlotDevicesInformation.cs
- host.json
- local.settings.json

Actions

Manage in Cloud Explorer

Manage Azure App Service settings

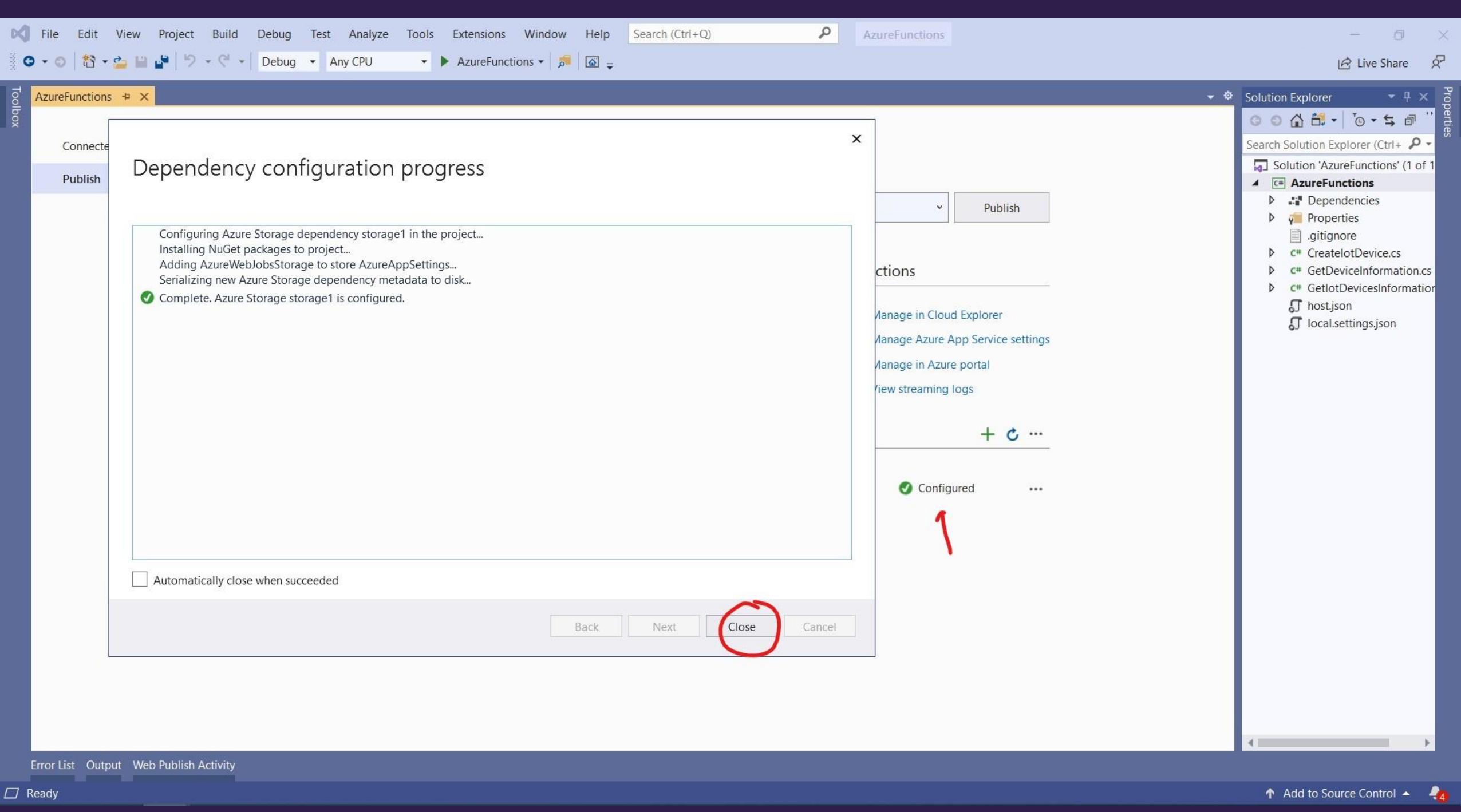
Manage in Azure portal

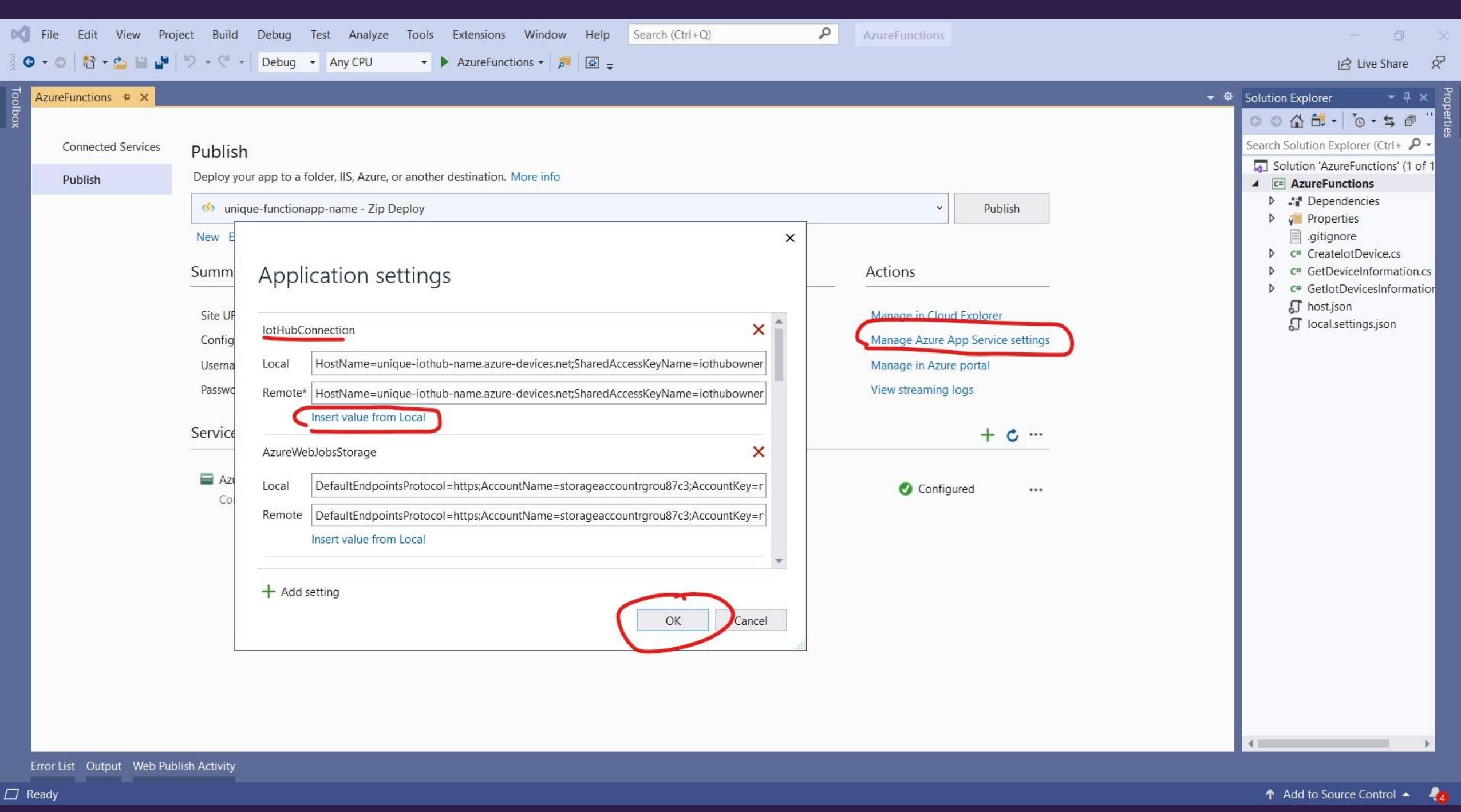
View streaming logs

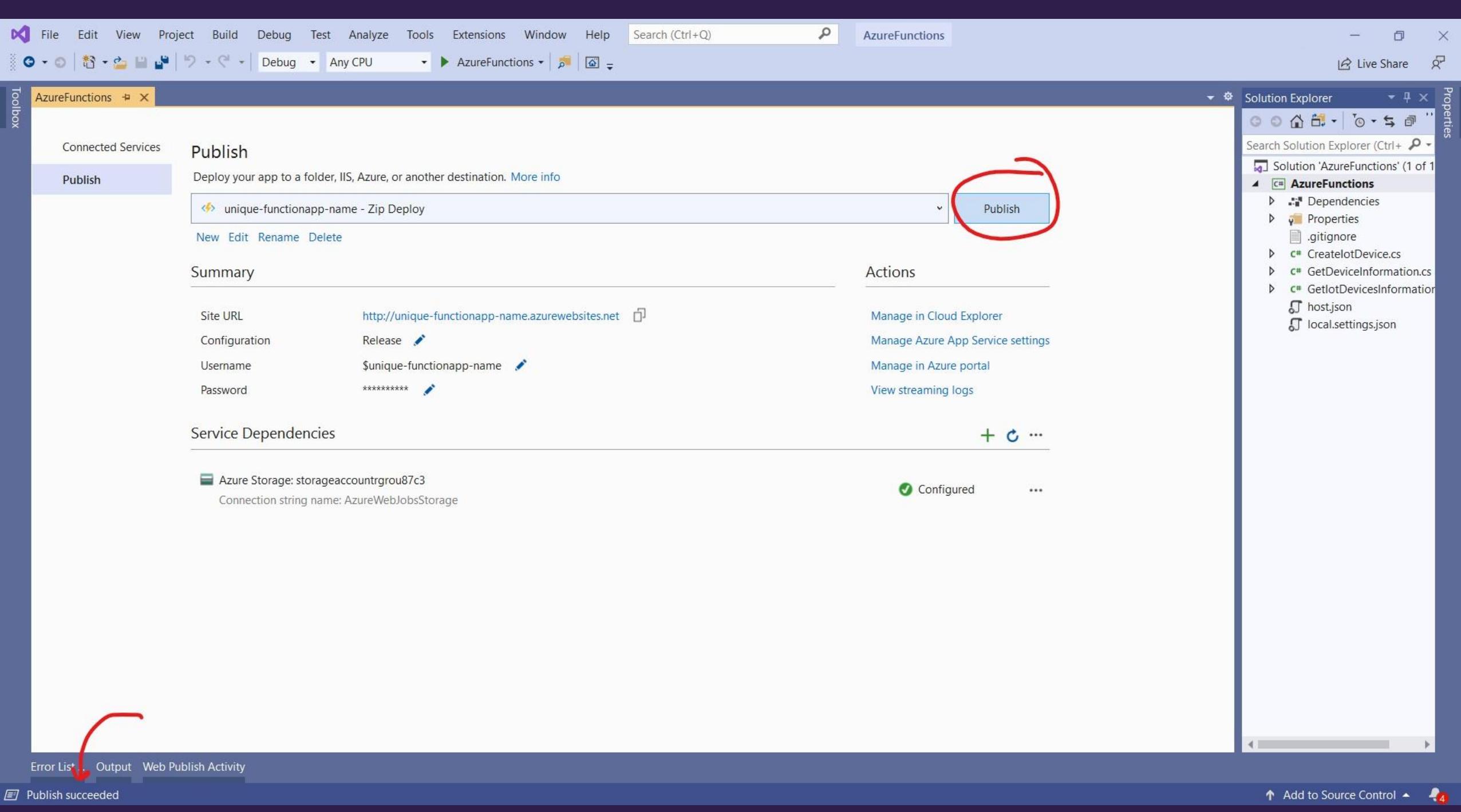
+

Configure

Error List Output Web Publish Activity







unique-functionapp-name | Func X +

https://portal.azure.com/#@34e9a1f3-23e1-4ead-b2fd-41660c25cc47/resource/subscriptions/881b6d88-33ad-42ff-a833-9d8630... ☆ 1 ⓘ 2 ⓘ ? ⓘ

Microsoft Azure

Home >

{fx} unique-functionapp-name | Functions

App Service

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Security

Events (preview)

Functions

Functions (highlighted with a red box)

App keys

App files

Proxies

Deployment

Deployment slots

Deployment Center

Settings

Configuration

Your app is currently in read only mode because you are running from a package file. To make any changes update the content in your zip file and WEBSITE_RUN_FROM_PACKAGE app setting.

+ Add Develop Locally Refresh Enable Disable Delete

Filter by name...

Name ↑	Trigger ↑	Status ↑
CreatelotDevice	HTTP	Enabled
GetDeviceInformation	HTTP	Enabled
GetlotDevicesInformation	HTTP	Enabled

A red arrow points upwards from the 'Functions' link in the left sidebar towards the 'Functions' section in the main content area.

GetlotDevicesInformation - Microsoft Edge

https://portal.azure.com/#blade/WebsitesExtension/FunctionMenuBlade/functionOverview/resourceId/%2Fsubscriptions%2F881... 1

Microsoft Azure

Home > unique-functionapp-name | Functions >

GetlotDevicesInformation

Function

Search (Ctrl+ /) | Enable | Disable | Delete | Get Function Url | Refresh

Get Function Url

default (function key) | unique-functionapp-name.azurewebsites.net/api/GetlotDevicesInformation? | OK

Developer

Code + Test | Integration | Monitor | Function Keys

Overview

Total Execution Count

Successful Execution Count

2 AM 2:15 AM 2:30 AM 2:45 AM UTC+02:00

2 AM 2:15 AM 2:30 AM 2:45 AM UTC+02:00

The screenshot shows the Azure Functions overview page for a function named 'GetlotDevicesInformation'. At the top, there's a toolbar with buttons for 'Enable', 'Disable', 'Delete', and 'Get Function Url'. The 'Get Function Url' button is circled in red. Below it, a modal dialog is open with the title 'Get Function Url'. It contains a dropdown menu showing 'default (function key)' and a text input field containing the URL 'unique-functionapp-name.azurewebsites.net/api/GetlotDevicesInformation?'. An 'OK' button is at the bottom of the dialog, with a red arrow pointing to it. On the left side, there's a sidebar with icons for 'Code + Test', 'Integration', 'Monitor', and 'Function Keys'. The main area has two charts: 'Total Execution Count' and 'Successful Execution Count', both showing execution counts over time from 2 AM to 2:45 AM UTC+02:00.