# Connect Arduino D1 (ESP8266) to Azure IoT Cloud

Create an Azure Cloud following steps here: <https://azure.microsoft.com/da-dk/documentation/samples/iot-hub-c-thingdev-getstartedkit/>

We use an Arduino D1 (Release 1) instead of the Sparkfun. The D1 is (I guess) an Uno with an ESP8266 built in.

## Create IOT Hub on Azure

<https://azure.microsoft.com/en-us/documentation/articles/iot-hub-csharp-csharp-getstarted/>

## IOT hub data (keys and connection strings)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Entity type | Entity name | Prim. key | Sec. key | Conn string prim key | Conn string sec key |
| IOT hub | NNRiothub.azure-devices.net | I5gHD6laJhVsYI6Bxz+33y1RuQsqLEJ0yZbTiXp2dBo= | ogb2DyDxpXzFT5y7HeB9OFXchbv1d0e1ASlIHUeppb0= | HostName=NNRiothub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=I5gHD6laJhVsYI6Bxz+33y1RuQsqLEJ0yZbTiXp2dBo= | : HostName=NNRiothub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=ogb2DyDxpXzFT5y7HeB9OFXchbv1d0e1ASlIHUeppb0= |
| Storage account | nnriothubstorage | Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g== | kshk0UuvpgRZRfcquABUmzBoDH0G3hmgrf0KHwVC6cX8RbVLSkGG/3KiYlHYeDXvyu2oSbSwHHLHDQc/w+2glA== | DefaultEndpointsProtocol=https;AccountName=nnriothubstorage;AccountKey=Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g==; | DefaultEndpointsProtocol=https;AccountName=nnriothubstorage;AccountKey=kshk0UuvpgRZRfcquABUmzBoDH0G3hmgrf0KHwVC6cX8RbVLSkGG/3KiYlHYeDXvyu2oSbSwHHLHDQc/w+2glA==; |
| Storage account access | nnraccess | URL:  https://nnriothubstorage.blob.core.windows.net/nnriothubcontainer?sr=c&sv=2015-12-11&si=nnraccess&spr=https&sig=3B%2BAbyfYUjRpsRri4iAVBhxUrtFSjfDiwa68814sepc%3D | SAS Token: **?sr=c&sv=2015-12-11&si=nnraccess&spr=https&sig=3B%2BAbyfYUjRpsRri4iAVBhxUrtFSjfDiwa68814sepc%3D**  **?sr=c&?sv=2015-12-11& si=nnraccess&spr=https&sig=zVwL9qrhi8LLQG6IQCVNltyvKcPCf%2BHbUQhsUjvAA34%3D**  **Generated 4/1-17:**  **?sv=2015-12-11&ss=bfqt&srt=sco&sp=rwdlacup&se=2017-01-24T23:46:16Z&st=2017-01-04T14:46:16Z&spr=https&sig=zVwL9qrhi8LLQG6IQCVNltyvKcPCf%2BHbUQhsUjvAA34%3D** |  | C# Code:  //You will need at least version 2.0 of the Azure sdk //The url of the blob service endpoint var blobEndPoint = new Uri("<your blob service endpoint>"); // The generated SAS token var sasToken = "?sr=c&sv=2015-12-11&si=nnraccess&spr=https&sig=3B%2BAbyfYUjRpsRri4iAVBhxUrtFSjfDiwa68814sepc%3D" ; var sasCredentials = new StorageCredentials(sasToken); // Create the CloudBlobClient using the shared access signature // as the credentials var cloudBlobClient = new CloudBlobClient(blobEndPoint, sasCredentials); |
| Container | nnriothubcontainer |  |  |  |  |
| Service Bus | nnriotServiceBus  ( RootManageSharedAccessKey) | aupcE7wKgQsi/grff+5QNECsl76INSTswBFqQ9qYwgU= | qUs8tQz5vGCTKYfz/gex4fPl9Z5mWZth+drabPFGNh8= | Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=aupcE7wKgQsi/grff+5QNECsl76INSTswBFqQ9qYwgU= | Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=qUs8tQz5vGCTKYfz/gex4fPl9Z5mWZth+drabPFGNh8= |
| Service Bus policy | nnriotServiceBus :Listen | iA+C1vfmK/OSv9/Kc6fa2rr4y0UXMjYuJT+bW7j3Y2E= |  |  |  |
| Service Bus policy | nnriotServiceBus :Send | JF7vEF1DyrqgDSjfKMghYSaskkG0RTrad4Qlnp7Elos= |  |  |  |
| Service Bus policy | nnriotServiceBus :Manage | 34VFEFyPlHbpgQajPbLyAg9CQruMBxSOyGtqqIukHGQ= |  |  |  |
| Service Bus Queue | Imagequeue |  |  |  |  |
| Service Bus **Queue** policy | Imagequeue:send | aXngTV7rLq0Kt9WYHbzltrQ/1kFWWSsyCbqmzQbwY1w= | MayKOUQHG8Y/NJZMFNo5HKHTuukWvAiDEZUWnl5qqUE= | Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=send;SharedAccessKey=aXngTV7rLq0Kt9WYHbzltrQ/1kFWWSsyCbqmzQbwY1w=;EntityPath=imagequeue | Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=send;SharedAccessKey=MayKOUQHG8Y/NJZMFNo5HKHTuukWvAiDEZUWnl5qqUE=;EntityPath=imagequeue |
| Service Bus **Queue** policy | Imagequeue:listen | RsaqaYhiHs3D/JYhwrfTdwYOEP+2DgdokyHoOoNgZFg= | N7wKF4OwA60LOOT5Md7tzxfOkBsrLXcnv4kV1pOeJAw= | Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=listen;SharedAccessKey=RsaqaYhiHs3D/JYhwrfTdwYOEP+2DgdokyHoOoNgZFg=;EntityPath=imagequeue | Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=listen;SharedAccessKey=N7wKF4OwA60LOOT5Md7tzxfOkBsrLXcnv4kV1pOeJAw=;EntityPath=imagequeue |
| Endpoint | CriticalQueue |  |  |  |  |
| Routes | ImageArrival |  |  | endpointName:CriticalQueue, exceptionMessage:An AMQP error occurred (condition='amqp:unauthorized-access'). TrackingId:da25f1cdeb49431fa8368bacf3a137b0\_G11, SystemTracker:gateway6, Timestamp:1/5/2017 3:44:09 PM. If you contact a support representative please include this correlation identifier: c87c2267-bc51-4bdb-8498-0415cbd92b11, timestamp: 2017-01-05 15:44:12Z, errorcode: IH400116. |  |
| Device simulated from c program “SimulatedDevice” | myFirstDevice | pyj6UeKzF/xCS+pM1ypYLNeAyljVa/tLPcS8ei+87Fg= |  |  |  |
| Arduino D1 | NNRArduinoD1\_01 | VsbmhBfDGJL4YpWaP8EIeykWCWClU0uPWfCWTNuajiA= | kYWOGOEwJeb3wjlSUXN0egVPyGcZCD6Jvz8tEE9fKwc= | HostName=NNRiothub.azure-devices.net;DeviceId=NNRArduinoD1\_01;SharedAccessKey=VsbmhBfDGJL4YpWaP8EIeykWCWClU0uPWfCWTNuajiA= | HostName=NNRiothub.azure-devices.net;DeviceId=NNRArduinoD1\_01;SharedAccessKey=kYWOGOEwJeb3wjlSUXN0egVPyGcZCD6Jvz8tEE9fKwc= |
| Arduino D1 | ArduinoD1\_001 | elB/d4TY5poTH8PpWH88EbqB8FHaGWSVRQ+INnorYPc= | VieD1fK2VGpG+I4kFO1bH+wE0x5MKdJNal6qhdLAL78= |  |  |
| AzureWebJobsDashboard |  |  |  | DefaultEndpointsProtocol=https;AccountName=functiond01811df800e;AccountKey=zTTicZ2Rj35lS+hPYyP6TK9WF8mCDIGSGxP/6+uHMkSqB+2/KO+I0F2qfqUZgrdDiFznpkvT24IIB3H1BF+dww== |  |
| AzureWebJobsStorage |  |  |  | DefaultEndpointsProtocol=https;AccountName=functiond01811df800e;AccountKey=zTTicZ2Rj35lS+hPYyP6TK9WF8mCDIGSGxP/6+uHMkSqB+2/KO+I0F2qfqUZgrdDiFznpkvT24IIB3H1BF+dww== |  |
| Aspose Barcode <https://dashboard.aspose.cloud/#/apps> | App SID: | fca8d3dc-a821-493c-8417-740de1131a79 | App Key: | 06dcc50d89436afc82eee57b4856df67 |  |

#### IOT HUB

<https://portal.azure.com/?whr=live.com#resource/subscriptions/e678a72c-f502-4d57-9066-b6ac1a8dda26/resourceGroups/NNR_iot_resource_group/providers/Microsoft.Devices/IotHubs/NNRiothub/IotHubKeys>

* IOT host name: NNRiothub.azure-devices.net
* Primary key: I5gHD6laJhVsYI6Bxz+33y1RuQsqLEJ0yZbTiXp2dBo=
* Sec key: ogb2DyDxpXzFT5y7HeB9OFXchbv1d0e1ASlIHUeppb0=
* Conn string, prim key: HostName=NNRiothub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=I5gHD6laJhVsYI6Bxz+33y1RuQsqLEJ0yZbTiXp2dBo=
* Conn string, sec key: HostName=NNRiothub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=ogb2DyDxpXzFT5y7HeB9OFXchbv1d0e1ASlIHUeppb0=

Subscription service: F1 (free)

Link to the Azure Portal: <https://portal.azure.com/?whr=live.com#resource/subscriptions/e678a72c-f502-4d57-9066-b6ac1a8dda26/resourcegroups/NNR_iot_resource_group/providers/Microsoft.Devices/IotHubs/NNRiothub/Overview>

#### Storage Account

<https://portal.azure.com/?whr=live.com#resource/subscriptions/e678a72c-f502-4d57-9066-b6ac1a8dda26/resourceGroups/NNR_iot_resource_group/providers/Microsoft.Storage/storageAccounts/nnriothubstorage/keys>

* Storage account name: nnriothubstorage
* **Key1**: Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g==
* **Key2**: kshk0UuvpgRZRfcquABUmzBoDH0G3hmgrf0KHwVC6cX8RbVLSkGG/3KiYlHYeDXvyu2oSbSwHHLHDQc/w+2glA==

#### Service Bus and Service Bus Queue

* **Service Bus**: nnriotServiceBus
* Service Bus (top level) listen, send, manage policies
* Service Bus Queue: imagequeue
* Imagequeue “listen” policy:

**Imagequeue “send” policy:**

* Primary key: rHqgdz6fYBaib8/2kROjBPt2nWVpcU1fF0UsAaOO9+g=
* Sec key: dbQdAttYEkH5cODPR3Q093e18UMeqdHgqyFAWv6hIHk=
* Conn str prim key: Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=send;SharedAccessKey=rHqgdz6fYBaib8/2kROjBPt2nWVpcU1fF0UsAaOO9+g=;EntityPath=imagequeue
* Conn str sec key: Endpoint=sb://nnriotservicebus.servicebus.windows.net/;SharedAccessKeyName=send;SharedAccessKey=dbQdAttYEkH5cODPR3Q093e18UMeqdHgqyFAWv6hIHk=;EntityPath=imagequeue

## Add device (Create a device identity)

There are several ways to create devices:

### Create device using the Azure portal

<https://azure.microsoft.com/en-us/documentation/articles/iot-hub-csharp-csharp-getstarted/>

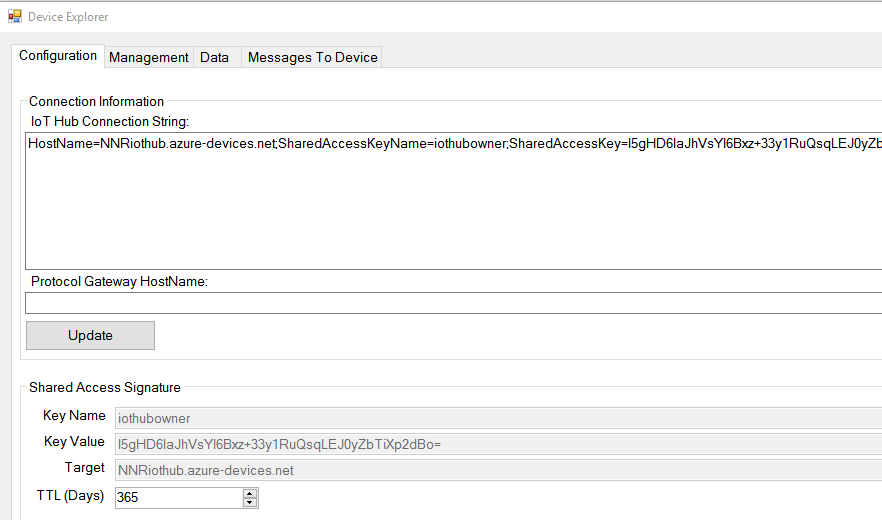
Generated device key:

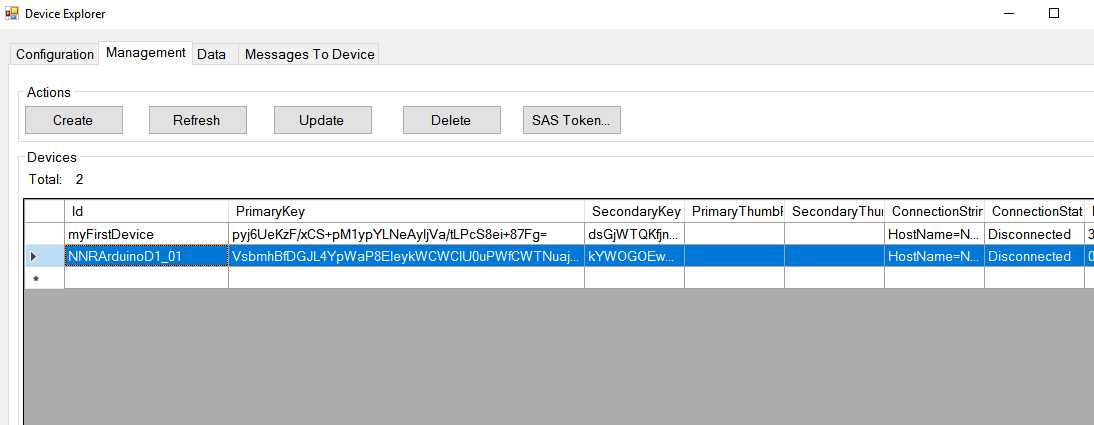
(follow the guide and create both a simulated device and a receiver).

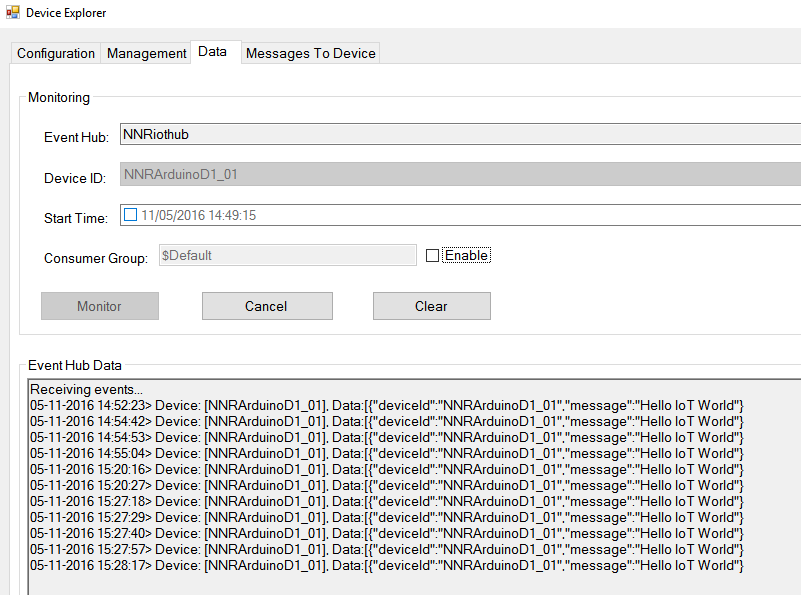
Then when that works, re-create the code on a real device: <https://azure.microsoft.com/da-dk/develop/iot/>

### Create device using a C-program on Windows PC

### Create device using the Windows program DeviceManager







## Arduino

### Sketch connecting to Azure Hub

Link: <https://github.com/Azure/azure-iot-sdks/blob/master/doc/manage_iot_hub.md>

Sketch: [Folder](file:///I:\DEV\GIT\Azure\ArduinoIoTHubNodePing\devicesample\ESP8266\iothub_ping)

Change line 10+11 in iothub\_ping.cpp:

static const char\* deviceId = "NNRArduinoD1\_01";

static const char\* connectionString = "HostName=NNRiothub.azure-devices.net;DeviceId=NNRArduinoD1\_01;SharedAccessKey=VsbmhBfDGJL4YpWaP8EIeykWCWClU0uPWfCWTNuajiA=";

Change 2 lines in iothub\_ping sketch:

static char ssid[] = "nohrTDC";

static char pass[] = "<password>";

### Simple HTTP test

Link: <https://github.com/Azure-Samples/iot-hub-node-ping/tree/master/devicesample/ESP8266>

#### Error undefined reference setup

Error message when comipiling from VS:

core.a(core\_esp8266\_main.cpp.o)\*: (.text.\_ZL12loop\_wrapperv+0x4): undefined reference to setup

core.a(core\_esp8266\_main.cpp.o)\*: (.text.\_ZL12loop\_wrapperv+0x8): undefined reference to loop

core.a(core\_esp8266\_main.cpp.o)\*: In function loop\_wrapper

core\_esp8266\_main.cpp:56: undefined reference to setup

core\_esp8266\_main.cpp:56: undefined reference to loop

collect2.exe\*: error: ld returned 1 exit status

Did this, with no effect:

* Include libraries via VS/microVisual menu: ESP8266Wifi

## Arduino D1 ESP

### Simple test on Azure IOT Hub

For Visual Studio setup and simple test, follow this carefully:   
<https://msdn.microsoft.com/en-us/magazine/mt694088.aspx>

Note that it also contains some pretty useful explanations for later use, e.g. on use of JSON.

A few notes to the procedure:

* ***Instead*** of executing the iothub-explorer commands from a cmd prompt, run the windows application Device Explorer.

## Connect Beaglebone

Link: <https://azure.microsoft.com/da-dk/develop/iot/get-started/>

* Device ID: BBB\_01\_NNR
* IoT Hub Hostname: HelloHubNNR.azure-devices.net
* Device Key: 1gLeo8D2CvGDelCdEpSoig==

#### Set up Azure SDK

Links:

<https://github.com/Azure/azure-iot-sdks/blob/master/doc/get_started/debian-beaglebone-black-c.md>   
<https://github.com/Azure/azure-iot-sdks/issues/273>

Boiling down to:

* sudo apt-get update
* sudo apt-get install -y curl libcurl4-openssl-dev build-essential cmake git
* git clone --recursive <https://github.com/azure/azure-iot-sdks.git>
* apt-get install libssl-dev
* apt-get install uuid
* ./azure-iot-sdks/c/build\_all/linux/build.sh

# Send image from device to Azure and process the image

<https://azure.microsoft.com/en-us/documentation/articles/iot-hub-csharp-csharp-process-d2c/>

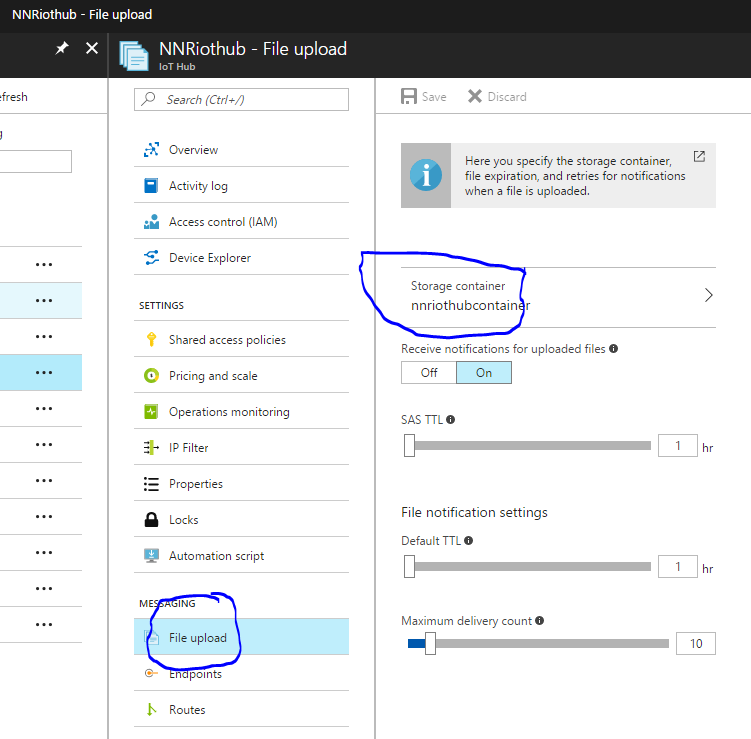
## Set up Azure

Create storage: Beginning of <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-csharp-csharp-file-upload/> (wait with the rest)

Enable File upload: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-configure-file-upload/>

Create container: <https://azure.microsoft.com/en-us/documentation/articles/iot-hub-configure-file-upload/>

Create/set: link between IOThub and storage container



Is this necessary:

Create Service Bus: **nnriotServiceBus**

Create Service Bus Queue: **imagequeue**

## Send the image

### Arduino program

Description of the base program used: <https://www.hackster.io/glovebox/secure-sensor-streaming-over-https-to-azure-iot-hub-dba05d>

How to build the HTTP request: <https://docs.microsoft.com/en-us/rest/api/storageservices/put-blob>

Code: <https://github.com/gloveboxes/Arduino-ESP8266-Secure-Azure-IoT-Hub-Client>

On top of this, I added:

* Code to fetch image from HTTP(S) server.
* Code to send to Azure Blob over wifi. The code chops up the data into smaller chunks to avoid memory overflow (Exception 29).
* Generate the SAS key to use on the Azure Portal in the storage resource. When generating the code, set the FROM-time back if you are not at UTC time. Otherwise you will get an error from the Azure server that the request is not within the time window.
* Use global variables for the large data sets to have stable memory allocations.

#### Less relevant info:

How to process IoT Hub device-to-cloud messages using .Net:   
<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-csharp-csharp-process-d2c>

How to upload files from devices to the cloud with IoT Hub:  
<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-csharp-csharp-file-upload>

If you get the error *("Message":"ErrorCode:InvalidStorageEndpointProperty;BadRequest")*, it’s because of **an error** in the Azure interface leading to a connection string not being set. Fix it using the guidance from   
<http://blogashwani.blogspot.dk/2016/08/file-upload-from-device-in-iot-hub.html>   
and the tool:  
<https://resources.azure.com/subscriptions/e678a72c-f502-4d57-9066-b6ac1a8dda26/resourceGroups/NNR_iot_resource_group/providers/Microsoft.Devices/IotHubs/NNRiothub>

The result will be: Allow a minute for Azure to update.

"storageEndpoints": {

"$default": {

"sasTtlAsIso8601": "PT1H",

"connectionString": "DefaultEndpointsProtocol=https;AccountName=nnriothubstorage;AccountKey=Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g==;",

"containerName": "nnriothubcontainer"

}

### Triggering the WebApp function from the Arduino

To get help in building the HTTP call bottom up (in Arduino), help can be found from the variable **webRequest** in SimulatedDevice. Use the watch functionality to debug the values of the HTTP call headers by looking at the exact values:

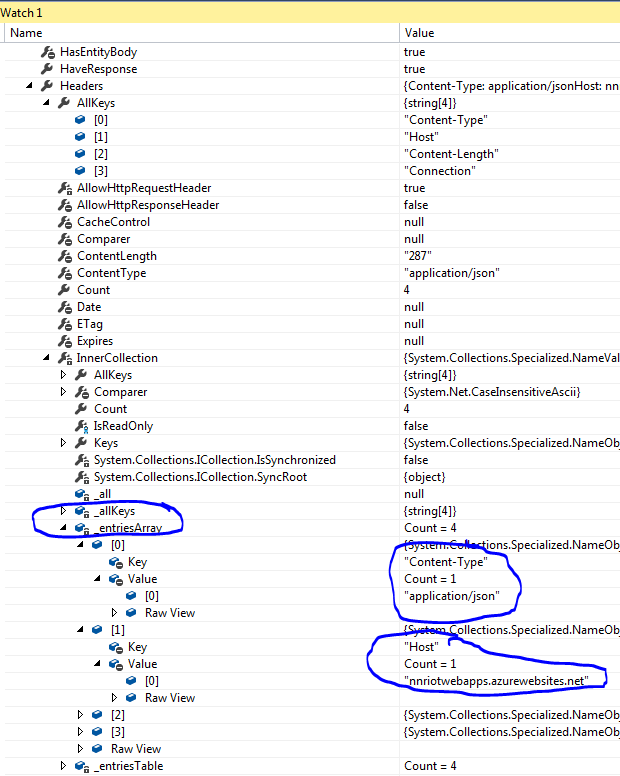


Figure 1. Screendump from the Visual Studio "watch" window of the variable webRequest.

## Process the image

Create a message queue according to <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-csharp-csharp-process-d2c>

The Service Bus Listen connection string is found here: <https://portal.azure.com/?whr=live.com#resource/subscriptions/e678a72c-f502-4d57-9066-b6ac1a8dda26/resourcegroups/nnr_iot_resource_group/providers/Microsoft.ServiceBus/namespaces/nnriotServiceBus/queues/imagequeue/saskey>

# Azure Web Functions and WebJobs (=”App Services”)

Excellent explanation of the difference between jobs and functions: <http://stackoverflow.com/questions/36610952/azure-webjobs-vs-azure-functions-how-to-choose>

This sample shows how to read/write from Storage from a .NET program (<https://azure.microsoft.com/da-dk/resources/samples/storage-blob-dotnet-getting-started/> ). This method is used from within the Web Function.

## Calling a WebAppFunction

### Building the HTTP request

Use PostMan app for Chrome to test. (<https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdggehcddcbncdddomop/related> )

#### Typical errors

|  |  |  |
| --- | --- | --- |
| Server reply | Potential error |  |
| <BODY><h2>Bad Request - Invalid Verb</h2> <hr><p>HTTP Error 400. The request verb is invalid.</p> | Header Content-Length too low. | Content-Length must match the body exactly INCL the surrounding brackets {}.  If in doubt, increase content-length. |
| (does not respond) | Header Content-Length too high. | The server expects more to come. |
|  |  |  |

## How to include a 3rd party API

Look for “Importing Namespaces” and “Package Management” in <https://docs.microsoft.com/en-us/azure/azure-functions/functions-reference-csharp> , i.e.:

* Add the project.json file to get the NuGet packagesw

### Haven OnDemand

APIKEY: 16b5dc6f-94f6-40eb-be96-62b7d49d7dc7

### Aspose Barcode for Cloud

<https://docs.aspose.com/display/barcodecloud/How+to+Run+the+Examples>

<https://github.com/aspose-barcode/Aspose.BarCode-for-Cloud/blob/master/Examples/DotNET/CSharp/ManagingRecognition/WithoutCloudStorage/ReadBarcodeFromLocalFile.cs>

NOTE: Works on the following **project.json** configuration. Has also been tested on v 1.0.5, and on net40 but does not work. Gives s String not defined error.

"frameworks": {

"net4**6**": {

"dependencies": {

"Aspose.BarCode-Cloud": "1.0.**3**"

}

},

"net40": {

"dependencies": {

"Newtonsoft.Json": "9.0.1"

}

}

}

}

# Azure Messaging

Examples: <https://github.com/Azure-Samples/azure-servicebus-messaging-samples>

## Setting up

From powershell:

PS C:\> Add-AzureAccount

Setup.ps1 does not really work. So create these entities from the Azure portal instead:

* Service bus => Shared access policies =>
* Listen, Send, manage (one policy for each)
* Enter the SAS keys in the azure-msg-config.properties file

The EntityPath is “imagequeue”, i.e. the queuename.

Get this to run: <https://github.com/Azure-Samples/azure-servicebus-messaging-samples/tree/master/QueuesGettingStarted>

# Visual Studio integration

## Install MS VS 2017

* Uninstall VS2015 COMPLETELY: <https://blogs.msdn.microsoft.com/heaths/2015/07/17/removing-visual-studio-components-left-behind-after-an-uninstall/> : From cmdline run:

en\_visual\_studio\_community\_2015\_with\_update\_3\_x86\_x64\_web\_installer\_8922963.exe /uninstall /force

* Install latest .NET: <https://www.microsoft.com/en-us/download/details.aspx?id=49982>
* Install VS 2017. In case of installation error, read the last line in the log file. It states the erroneous command. Re-execute it manually.   
  In my case:

Microsoft Windows [Version 6.1.7601]

Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\nnohrras>"c:\windows\syswow64\\windowspowershell\v1.0\powershell.exe" -NoLogo -NoProfile -ExecutionPolicy Unrestricted -InputFormat None -Command "& """C:\ProgramData\Microsoft\VisualStudio\P

ackages\Win10SDK\_10.0.14393.795,version=10.0.14393.79501\WinSdkInstall.ps1""" -SetupExe sdksetup.exe -SetupLogFolder standalonesdk -PackageId Win10SDK\_10.0.14393.795 -LogFile """C:\Users\nnohrras\AppD

ata\Local\Temp\dd\_setup\_20170310121620\_135\_Win10SDK\_10.0.14393.795.log""" -SetupParameters """/features OptionId.AvrfExternal OptionId.WindowsSoftwareDevelopmentKit OptionId.WindowsSoftwareLogoToolkit

OptionId.NetFxSoftwareDevelopmentKit /quiet /norestart""";

Removing existing target folder C:\Users\nnohrras\AppData\Local\Temp\Win10SDK\_10.0.14393.795.

Creating target folder C:\Users\nnohrras\AppData\Local\Temp\Win10SDK\_10.0.14393.795.

Launching 'sdksetup.exe' with arguments '/features OptionId.AvrfExternal OptionId.WindowsSoftwareDevelopmentKit OptionId.WindowsSoftwareLogoToolkit OptionId.NetFxSoftwareDevelopmentKit /quiet /norestart /Log "C:\Users\nnohrras\AppData\Local\Temp\standalonesdk\dd\_setup\_20170310121620\_135\_Win10SDK\_10.0.14393.795.sdksetup.log"'.

Waiting for child process to terminate.

Child process terminated with exit code 0.

* Un+re-install IIS Express (NOTE: This did not work, although I tried it):  
  (<https://developercommunity.visualstudio.com/content/problem/24464/packageidmicrosoftancmiisexpressmsipackageactionin.html> )
* MS Azure App Service Tools 2.9.6 for VC 201**5** (WebToolsAzure2015.exe)
* From VS tools menu, install:
  + Arduino IDE
* If more tools / applications needs to be installed, use the file vs\_community\_\_1790666033.1478185674.exe  
  Can also be opened from within VS: New project => “Not what you’re looking for” => “Open Visual Studio Installer”

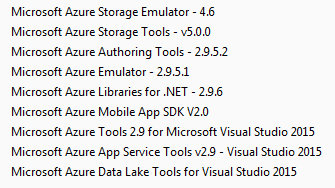
### Install Azure Functions (WebJobs)

(not needed for VS2017. It will be included:  
<https://buildazure.com/2016/12/14/azure-functions-visual-studio-tools-preview/>

## Install MS VS 2015

Install Visual Studio 2015 (takes a few hours).

Install

* WEB Developer (from Control Panel => Programs). This takes hours.
* Microsoft Azure SDK for .NET: <https://azure.microsoft.com/en-us/downloads/>   
  Which also installs:  
  
* NuGet package manager for VS
* Visual Micro Arduino Ide

## Install Azure Functions (WebJobs)

WebToolsAzureVS2015.exe: <https://blogs.msdn.microsoft.com/webdev/2016/12/01/visual-studio-tools-for-azure-functions/>

<https://docs.microsoft.com/en-us/azure/app-service-web/websites-dotnet-deploy-webjobs>

### To debug remotely (on Azure server) from Visual Studio

<https://docs.microsoft.com/en-us/azure/app-service-web/web-sites-dotnet-troubleshoot-visual-studio>

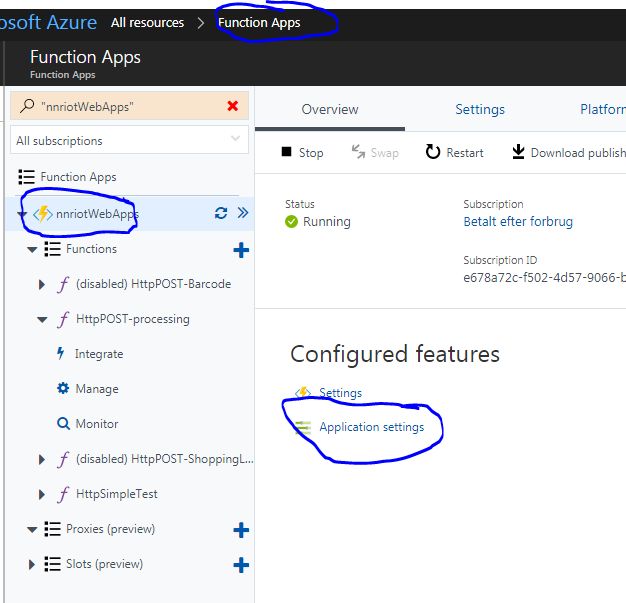
### to debug locally in Visual Studio:

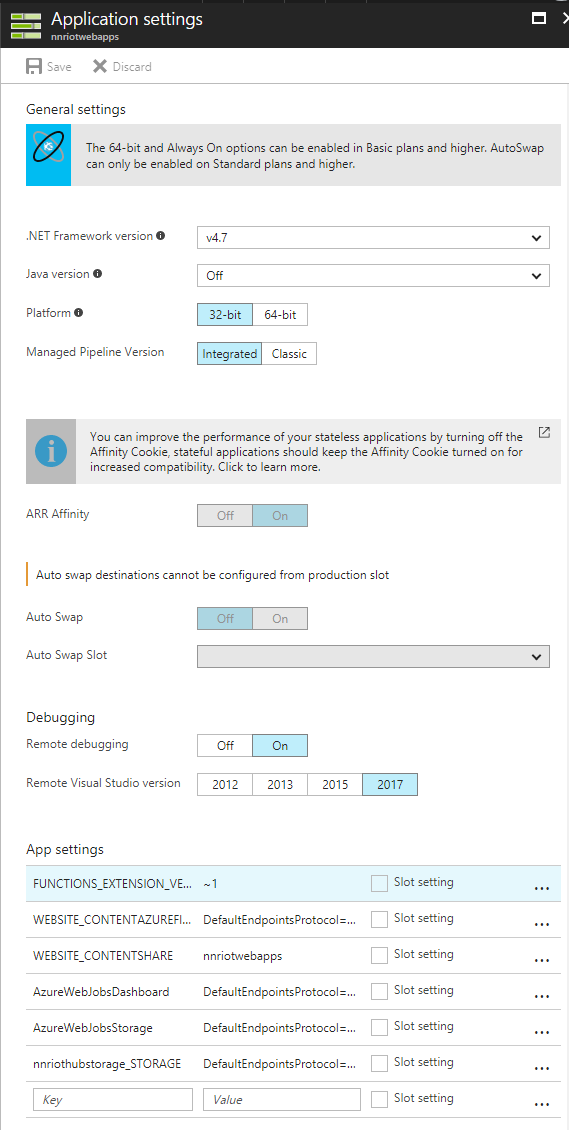
<https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/azure-functions/functions-run-local.md>

Do as follows:

 first [install NodeJS](https://docs.npmjs.com/getting-started/installing-node)

Create 5 env variables in Azure Function:





|  |  |
| --- | --- |
| WEBSITE\_CONTENTAZUREFILECONNECTIONSTRING | DefaultEndpointsProtocol=https;AccountName=nnriothubstorage;AccountKey=Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g== |
| WEBSITE\_CONTENTSHARE | nnriotwebapps |
| AzureWebJobsDashboard | DefaultEndpointsProtocol=https;AccountName=nnriothubstorage;AccountKey=Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g== |
| AzureWebJobsStorage | DefaultEndpointsProtocol=https;AccountName=nnriothubstorage;AccountKey=Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g== |
| nnriothubstorage\_STORAGE | DefaultEndpointsProtocol=https;AccountName=nnriothubstorage;AccountKey=Dwho3wxRlVaHbIgoQCuUSU0EjZlCunAdah3+JU7syboA4KCJoDjp+7KGI09rTRRRRSAre++FFR1WRbDFCfpc+g== |
|  |  |

## In case Azure won’t compile

It’s maybe a pointer/array error. Try editing directly from the browser. I did not find a 100% certain way through this problem.

## Install Cloud Explorer

## <https://marketplace.visualstudio.com/items?itemName=MicrosoftCloudExplorer.CloudExplorerforVisualStudio2015>

<https://docs.microsoft.com/en-us/azure/vs-azure-tools-resources-managing-with-cloud-explorer>

# Powershell

## Installation (non Windows10 users)

<https://docs.microsoft.com/da-dk/powershell/azureps-cmdlets-docs/>

* If you DON’T have Windows 10, install <https://www.powershellgallery.com/>.  
  (if you do, it’s already built-in).

# Other stuff

* Azure Storage connected service
* (Azure Power shell )
* Azure Cross-platform CLI
* [Microsoft.Azure.WebJobs](http://www.nuget.org/packages/Microsoft.Azure.WebJobs/)
* [Microsoft.Azure.WebJobs.Core](http://www.nuget.org/packages/Microsoft.Azure.WebJobs.Core/)
* Microsoft Azure Common Library
* [Microsoft Azure Service Bus](http://www.nuget.org/packages/WindowsAzure.ServiceBus/)
* [Microsoft.Azure.WebJobs.Extensions](http://www.nuget.org/packages/Microsoft.Azure.WebJobs.Extensions/)
* [Microsoft Azure Common Library Dependencies](http://www.nuget.org/packages/Microsoft.WindowsAzure.Common.Dependencies/)

Link for other Azure extensions/tools: <http://www.nuget.org/packages?q=windowsazureofficial>