## References

* Azure Blob Storage

<https://azure.microsoft.com/zh-tw/services/storage/blobs/>

* Get started with Azure Blob storage using .NET

<https://docs.microsoft.com/en-us/azure/storage/storage-dotnet-how-to-use-blobs>

* Service Bus

<https://azure.microsoft.com/zh-tw/services/service-bus/>

* Get started with Service Bus queues

<https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-dotnet-get-started-with-queues>

## Requirements1

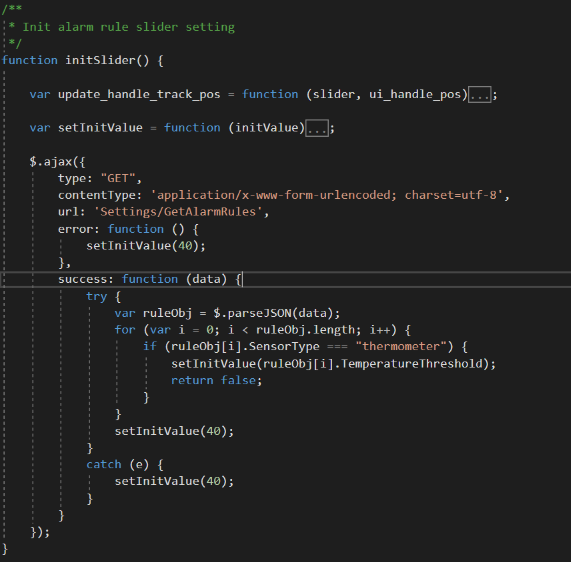
* Finished the 5th hands-on lab
* Device Simulator
* NuGet packages
  + WindowsAzure.ServiceBus for Service Bus
  + WindowsAzure.Storage for Blob storage access

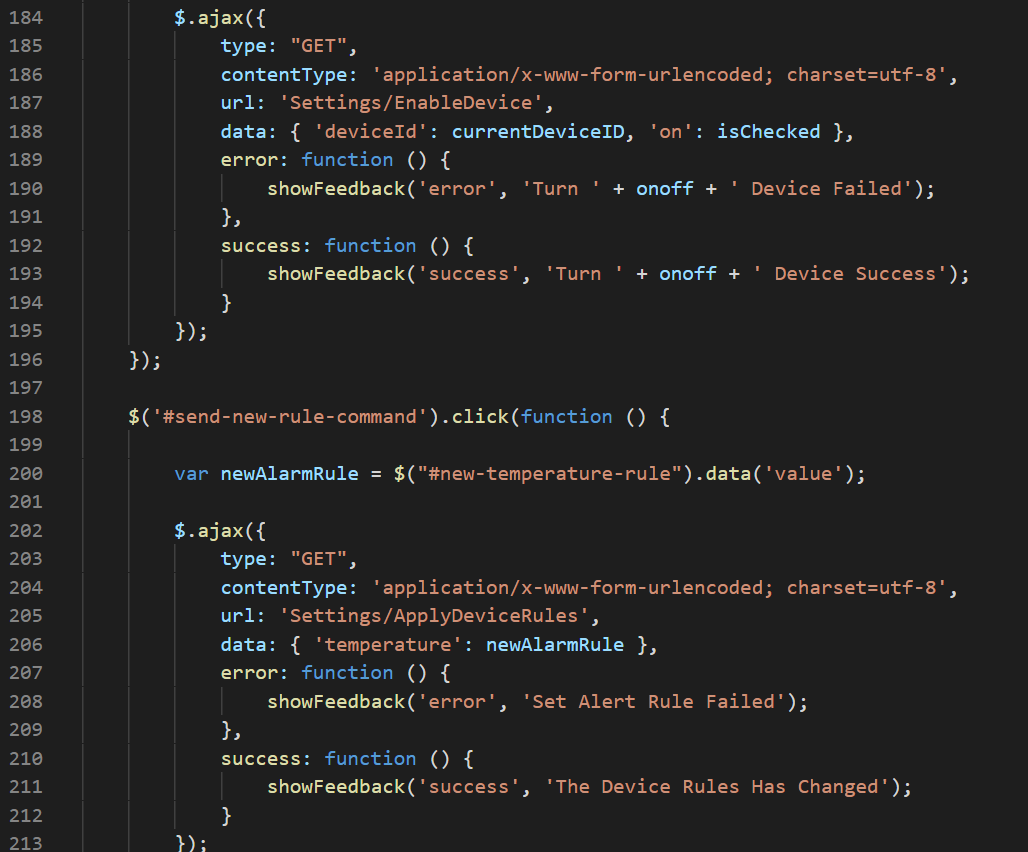
## Goals

* Set the device rules by dashboard
  + Update the reference Blob file.
* Control the device by dashboard
  + Send the alarm message to Alarm Service Bus
* Check the receiving commands of the simulated devices

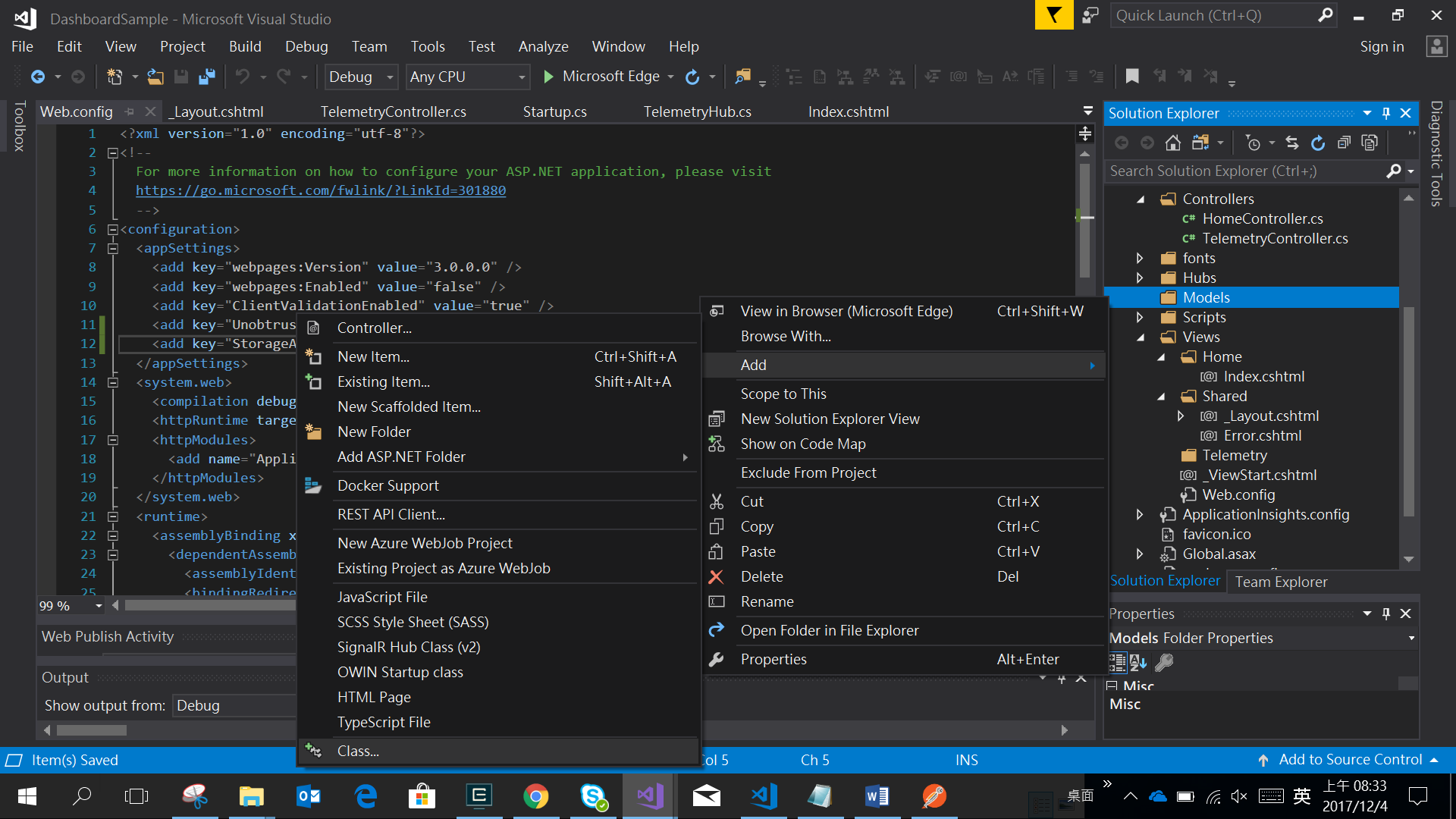
## Step 1: Add the Setting Controller to communication between the Web Server and ajax of Client

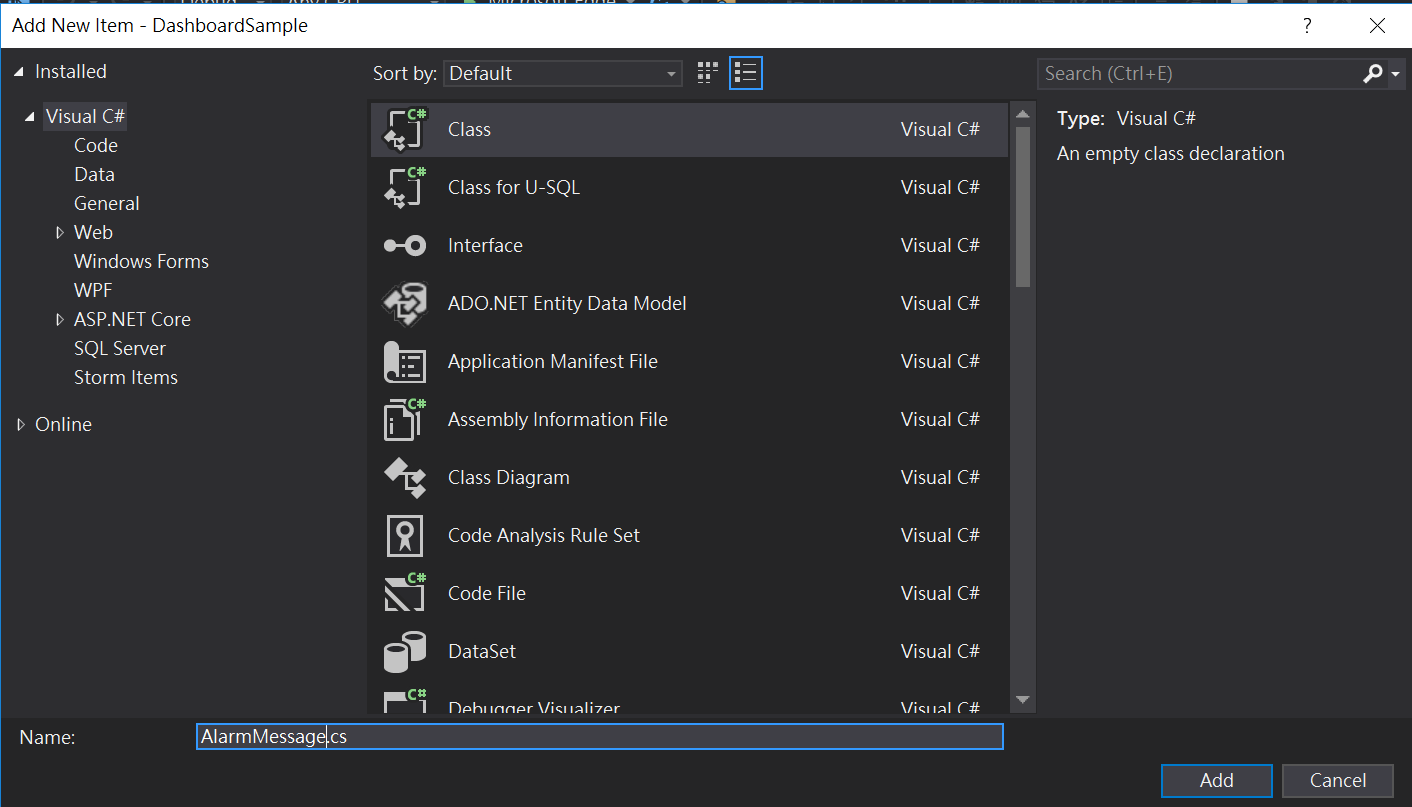
* View the ajax of **DashboardSample/Assets/js/widgetLayout.js, and we already created 3 handlers for**
  + url: 'Settings/GetAlarmRules'
  + url: 'Settings/EnableDevice'
  + url: 'Settings/ApplyDeviceRules'





* Add the **AlarmMessage.cs** Class in **Models**.





* Change the contents of the file to the following.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DashboardSample.Models

{

public class AlarmMessage

{

public string ioTHubDeviceID { get; set; }

public string alarmType { get; set; }

public string reading { get; set; }

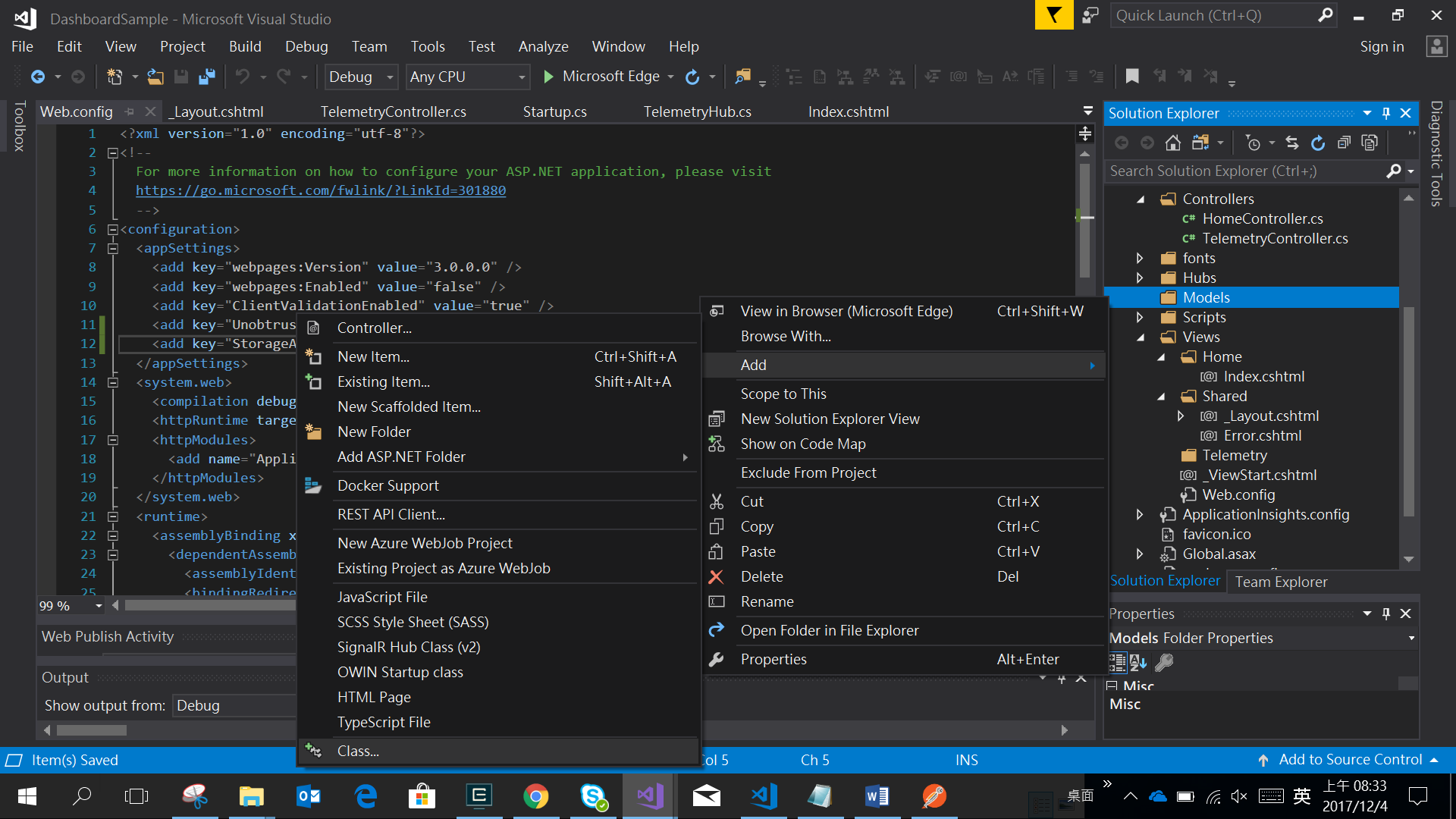
public string threshold { get; set; }

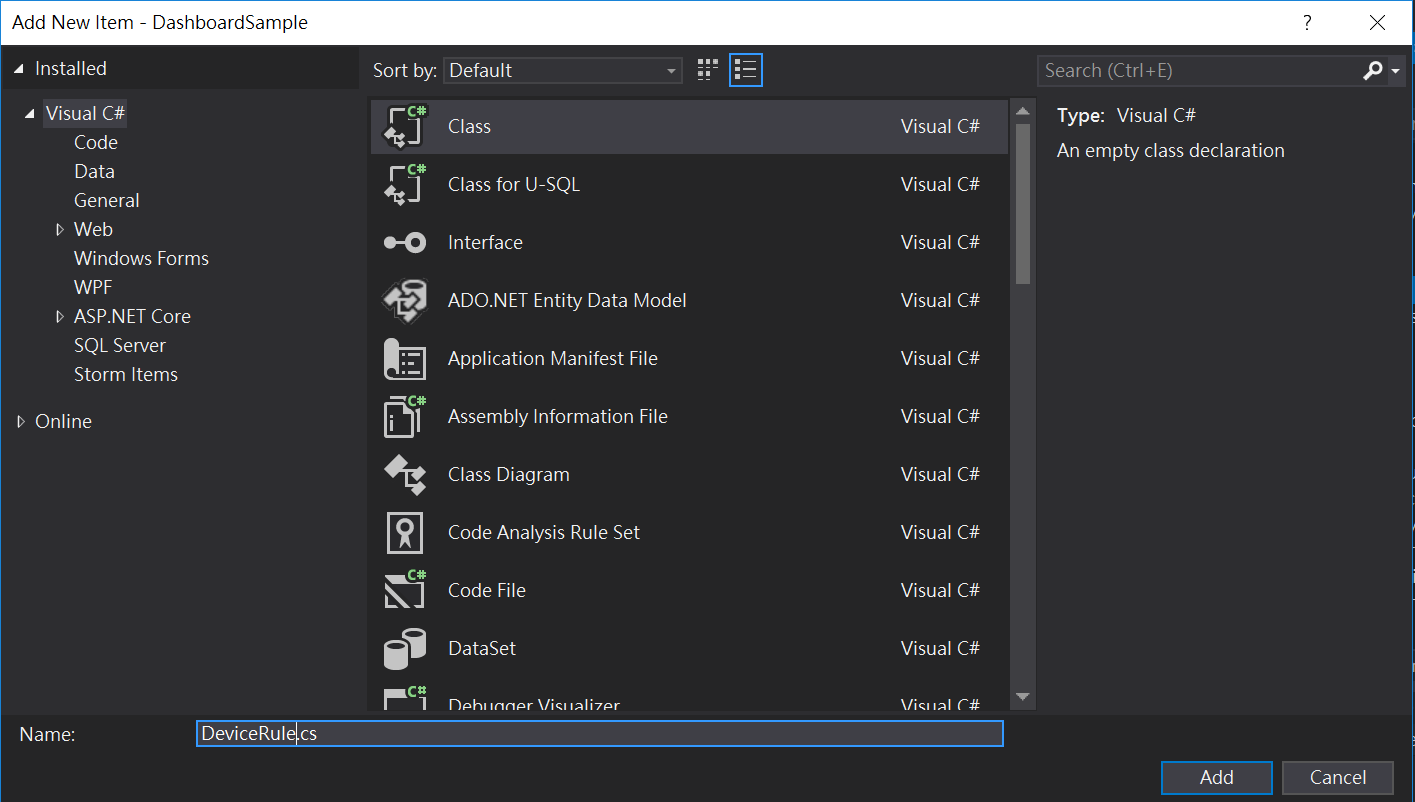
public string createdAt { get; set; }

}

}

* Add the **DeviceRule.cs** in **Models**.





* Change the content of the file to the following.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DashboardSample.Models

{

public class DeviceRule

{

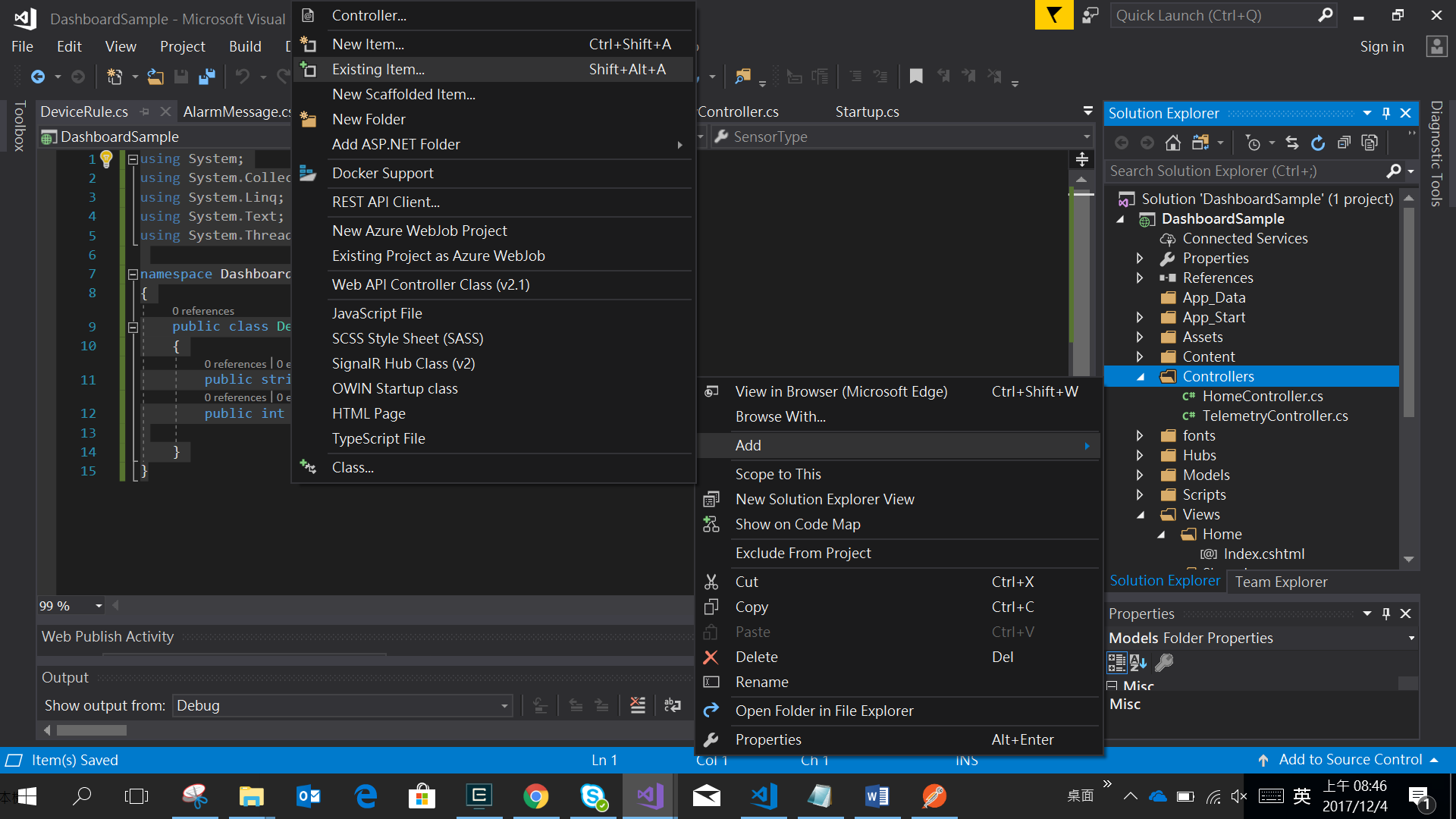
public string SensorType { get; set; }

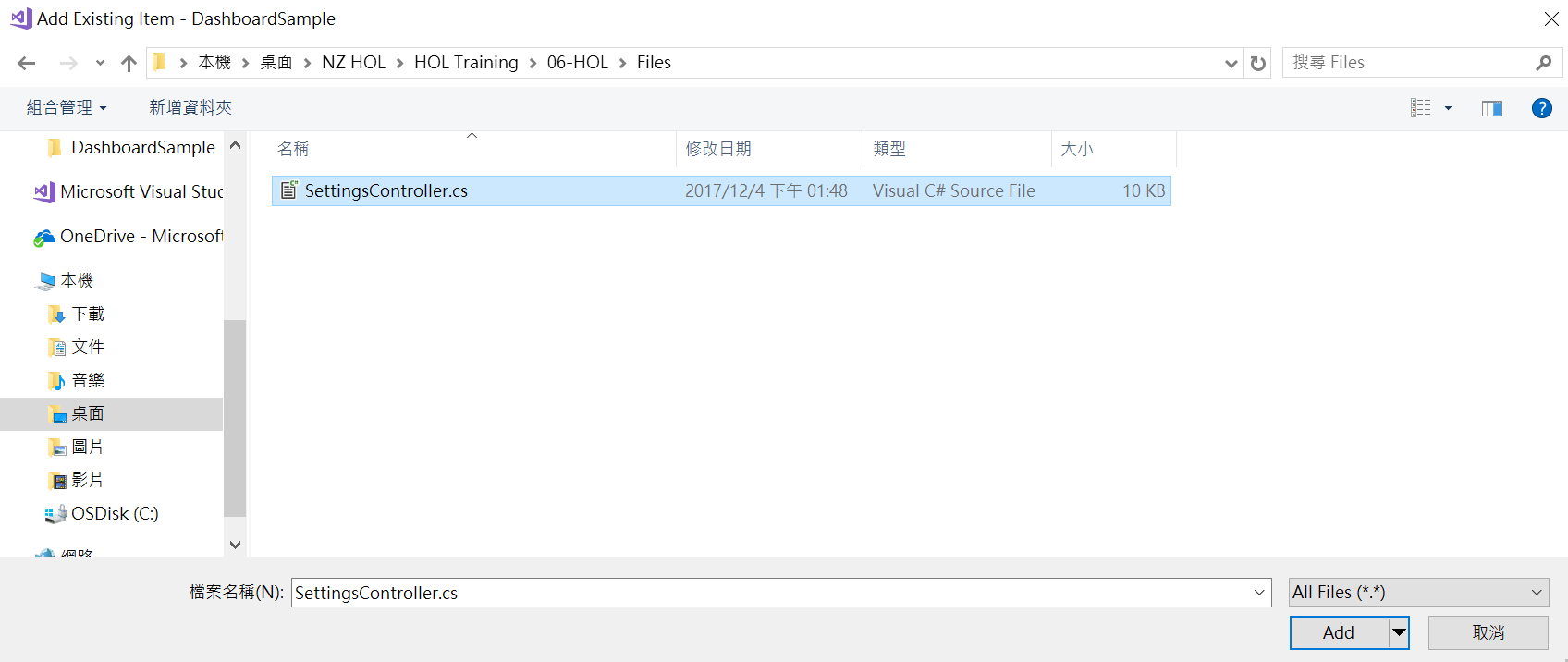
public int TemperatureThreshold { get; set; }

}

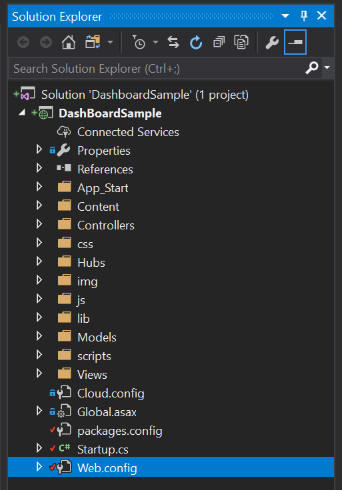
}

* Add the **06-HOL** >> **Files** >> **SettingsController.cs** file into **Controllers** folder.   
  (Edit the line, and add   
  using System.IO;   
  var message = new BrokeredMessage(new MemoryStream(Encoding.UTF8.GetBytes(msg)), true);  
  instead of var message = new BrokeredMessage(msg);)



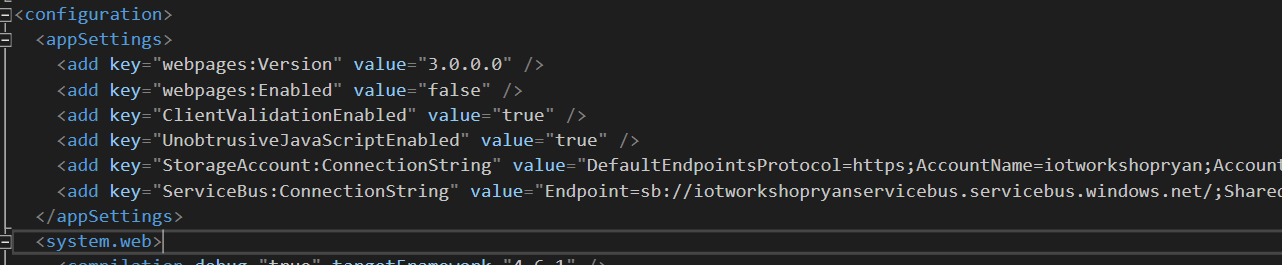


* Add the settings into **appSettings** of **Web.config** as the following.
  + **StorageAccount:ConnectionString**: the connection string of Storage Account
  + **ServiceBus:ConnectionString**: the connection string of Service Bus

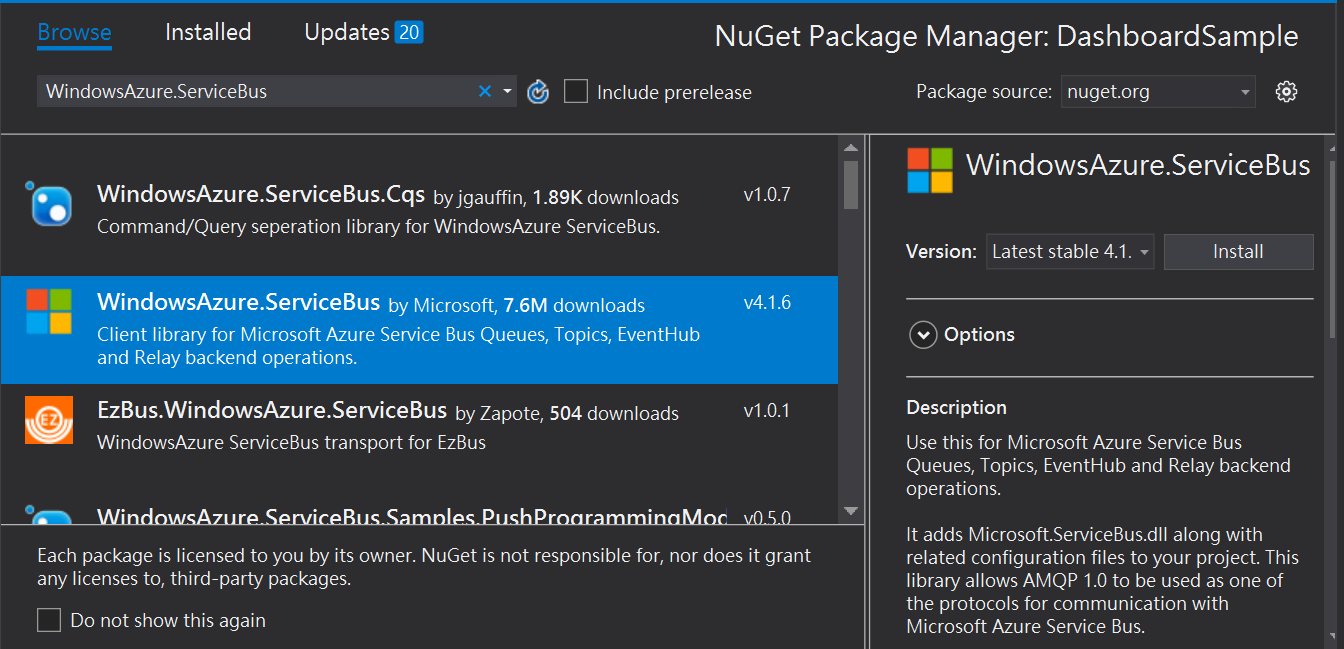


<add key="StorageAccount:ConnectionString" value="[Replace your Storage ConnectionString]" />

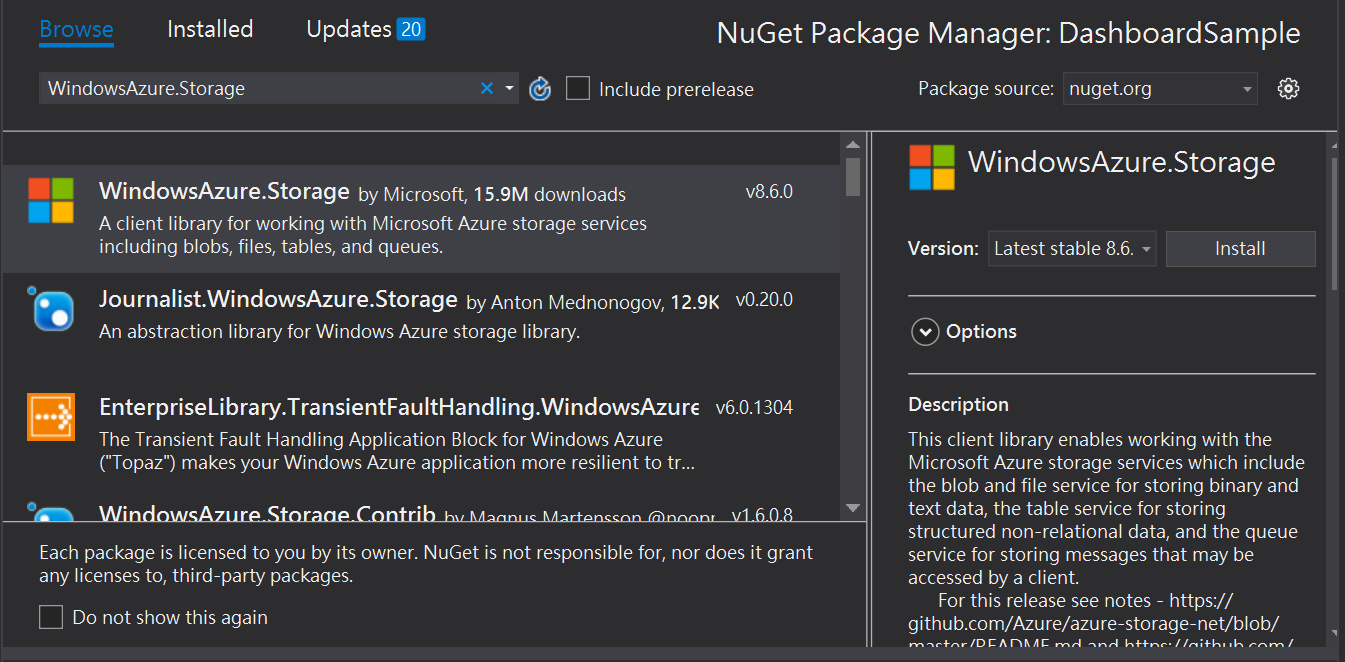
<add key="ServiceBus:ConnectionString" value="[Replace your ServiceBus ConnectionString]"/>



* Install the **WindowsAzure.ServiceBus** packagefor Service Bus, and click OK to accept Preview Changes and License Acceptance.

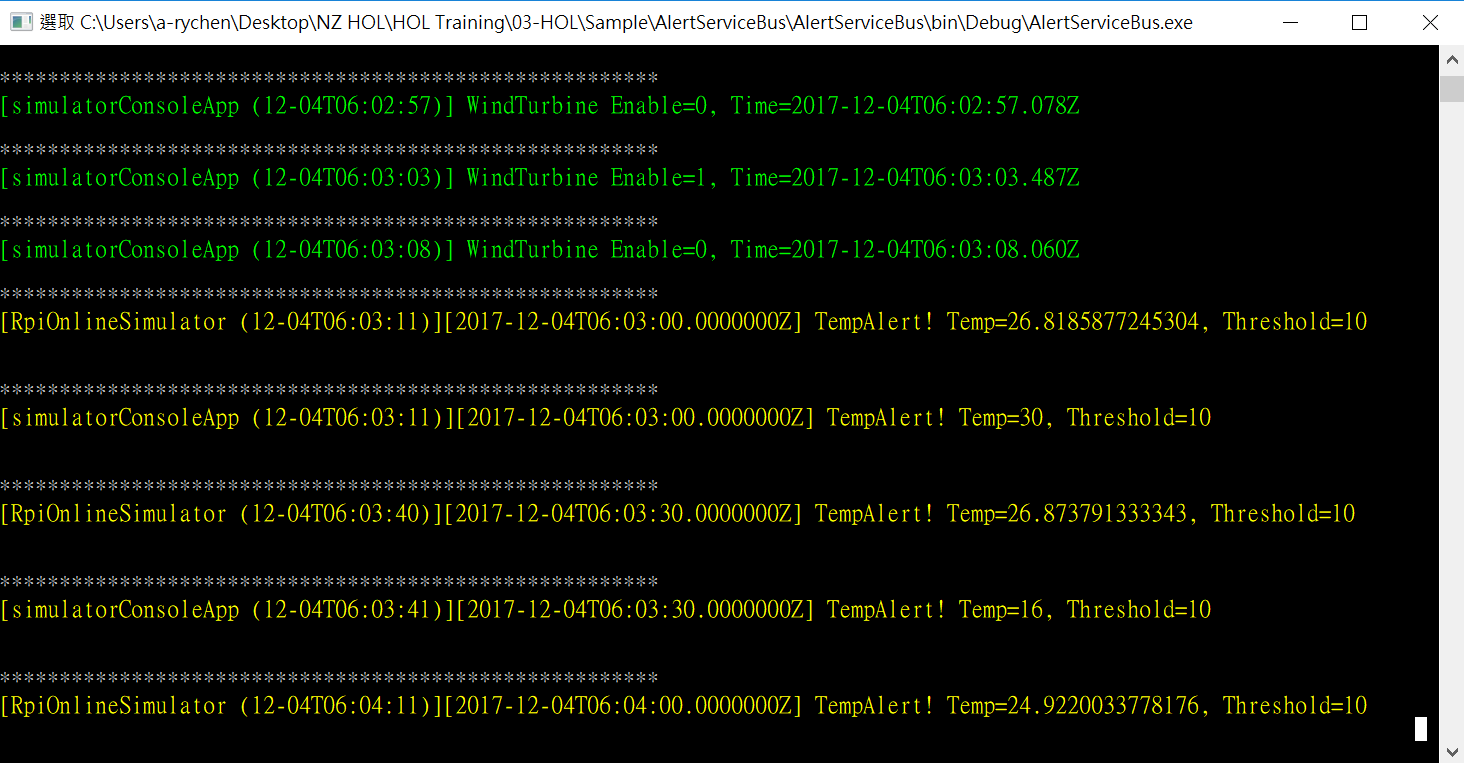


* Install the **WindowsAzure.Storage** packagefor Blob Storage, and click OK to accept Preview Changes and License Acceptance.

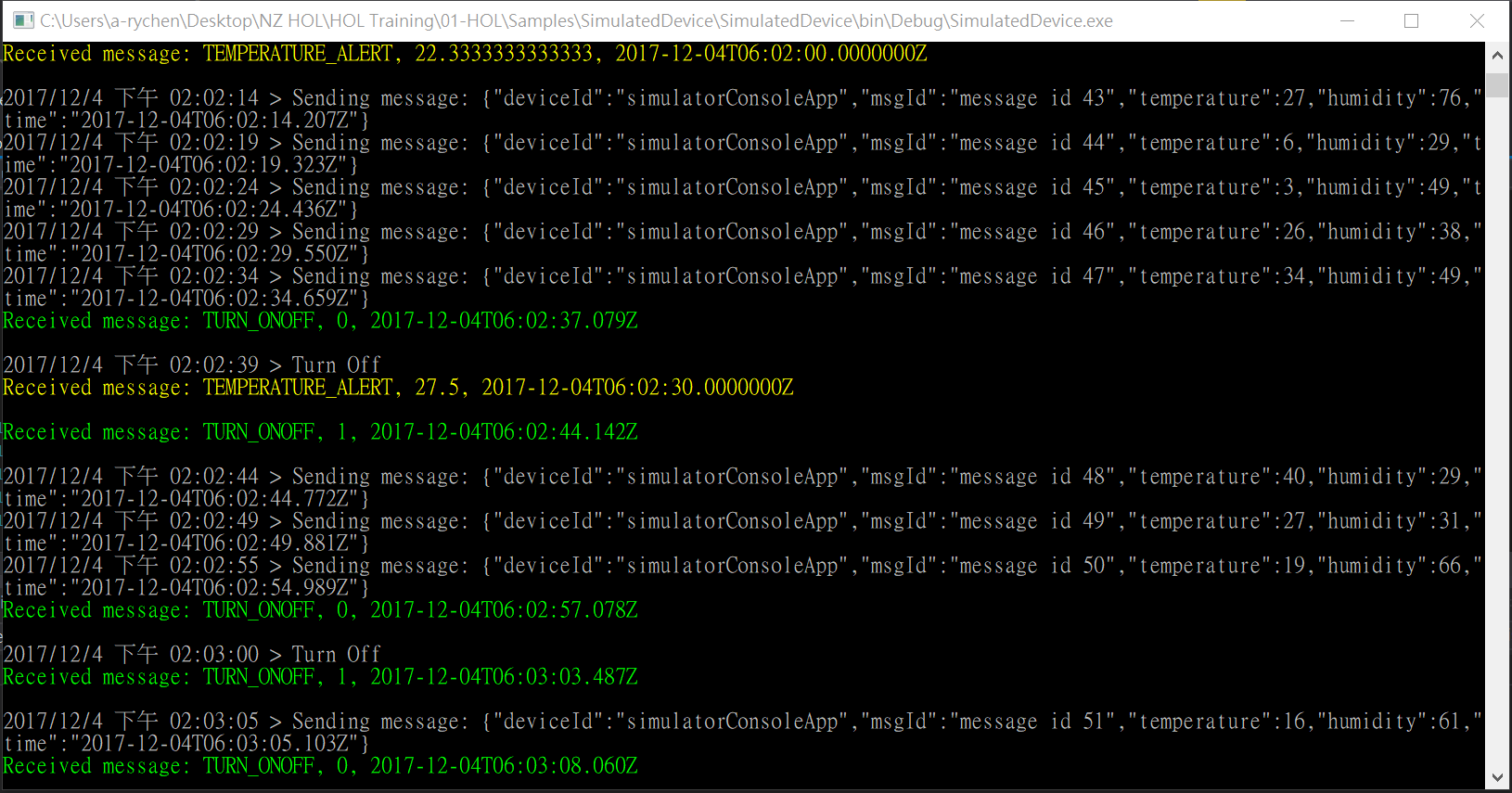


## Step 2: Run all applications

* Run the **Telemetry Event Processor Host** Console App.
* Run the **Alert Service Bus** Console App
* Run the **Device** **simulator.**
* Run the **DashboardSample Web App**, and you can
  + Control the ON/OFF of device
  + Set the alarm rules of device (will be affected after 2 minutes)
  + 
  + Watch the log of Alarm Service Bus



* + Check the C2D log of Device (**SimulatedDevice**)



* *The HOL 6 has been completed. Now You can get the full access to control the devices by dashboard, also configure the device rules that it should be triggered by Azure Stream Analytics in a few minutes.*