## 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>

## Requirements

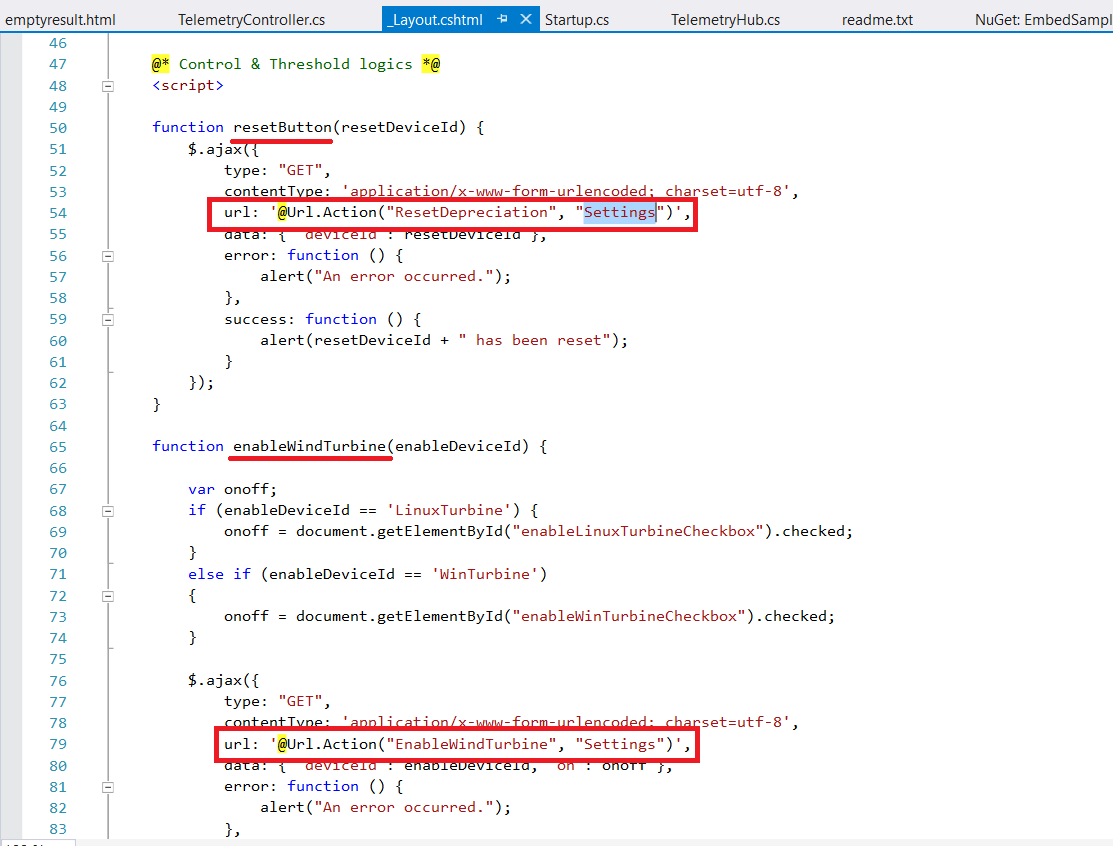
* Finished the part 10 of HOL
* Simulated Linux & Windows Wind turbines
* NuGet packages
  + Newtonsoft.Json for JSON in C#
  + WindowsAzure.ServiceBus for Service Bus
  + WindowsAzure.Storage for Blob storage access

## Goals

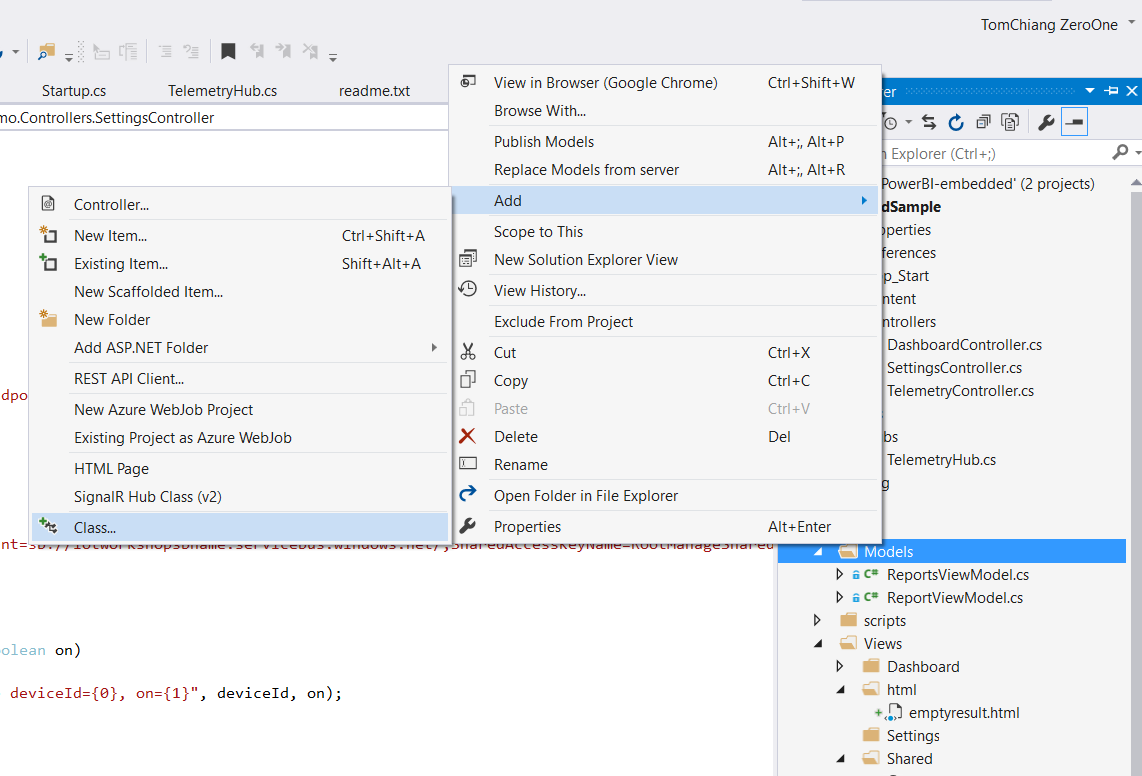
* Set the device rules by dashboard
  + Update the reference Blob file.
* Control the device by dashboard
  + Send the alarm message to Alarm Service Bus
* Check the receiving commands of the simulated devices

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

* View the ajax of **\_Layout.cshtml** file of EmbedSample project.



* 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 paas\_demo.Models

{

class AlarmMessage

{

public string ioTHubDeviceID { get; set; }

public string messageID { get; set; }

public string alarmType { get; set; }

public string reading { get; set; }

public string threshold { get; set; }

public string localTime { get; set; }

public string createdAt { get; set; }

}

}

* Add the **DeviceRule.cs** 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 paas\_demo.Models

{

class DeviceRule

{

public string DeviceID { get; set; }

public double CutOutSpeed { get; set; }

public double Repair { get; set; }

public double Altitude { get; set; }

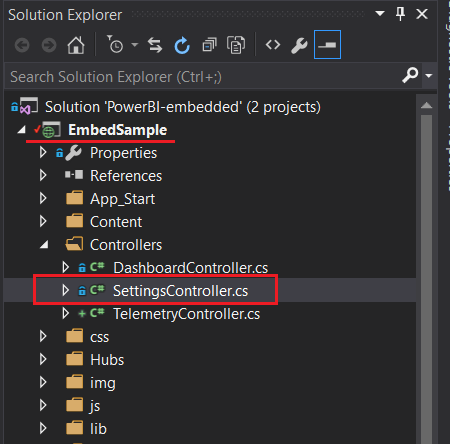
public double Latitude { get; set; }

public double Longitude { get; set; }

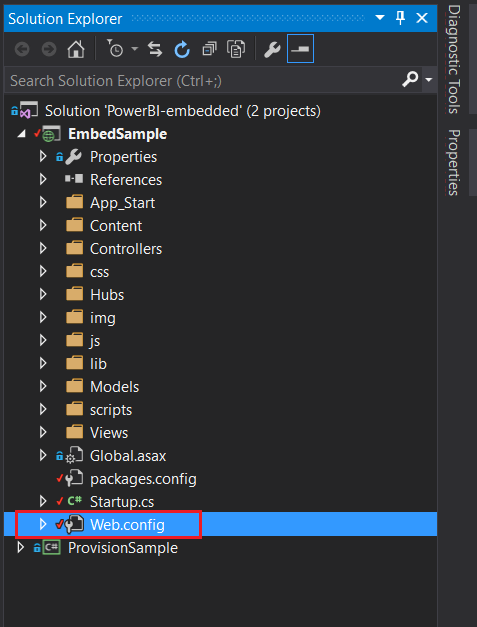
}

}

* Add the **SettingsController.cs** file in the **Controllers** folder.
  + Copy and paste the **SettingsController.cs** to this folder.



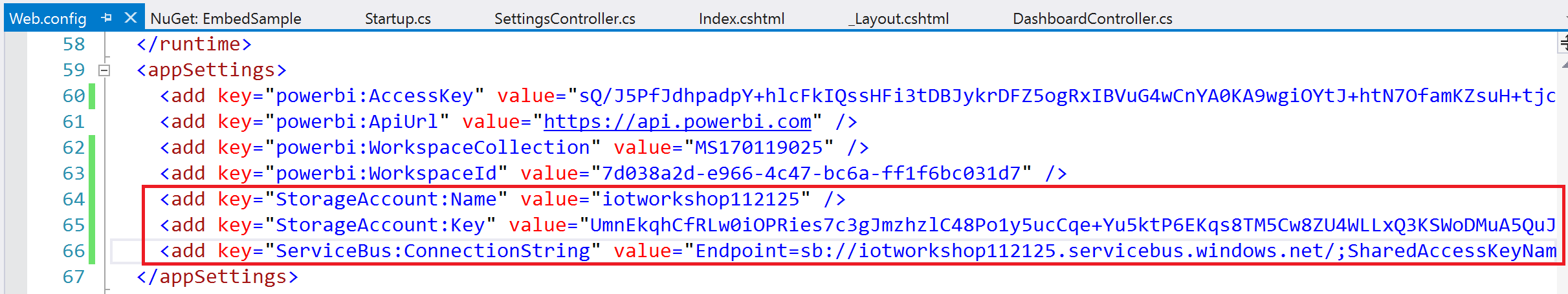
* Add the settins of **Web.config** as the following.
  + **StorageAccount:Name**: the name of Storage Account
  + **StorageAccount:Key**: the key of Storage Account
  + **ServiceBus:ConnectionString**: the connection string of Service Bus



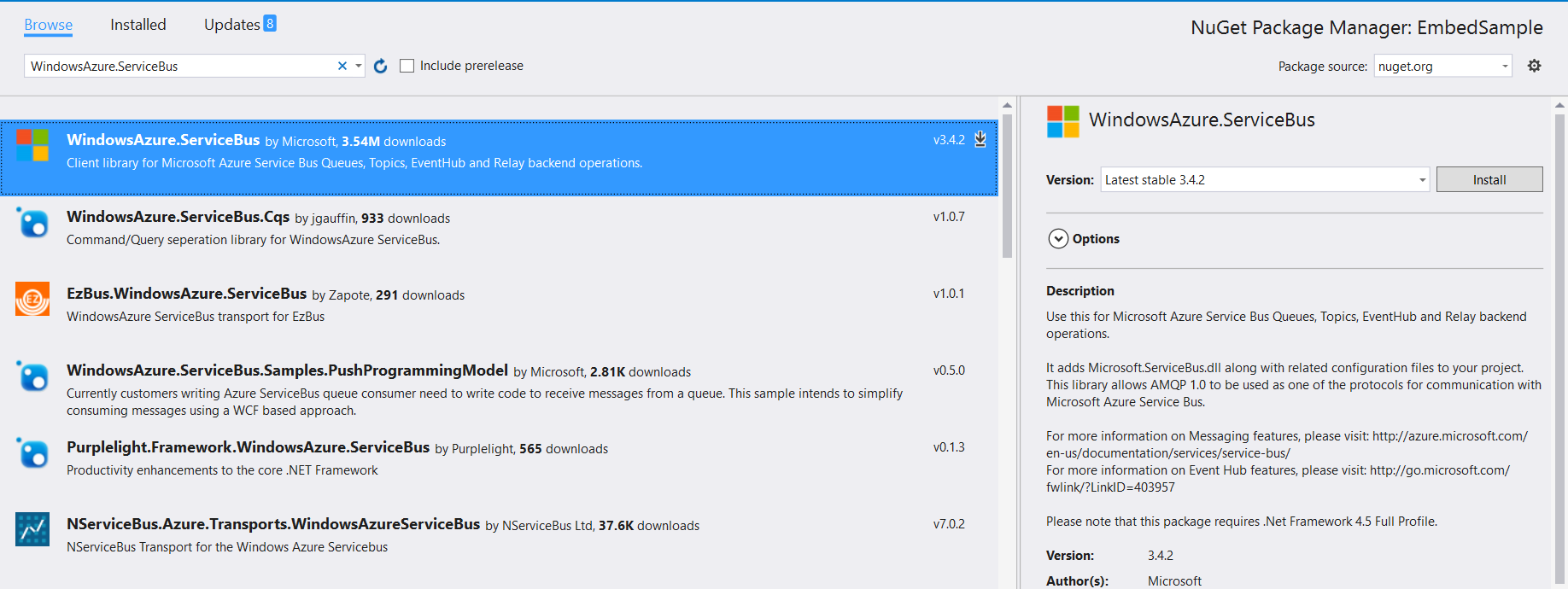
<add key="StorageAccount:Name" value="Replace your Storage account name" />

<add key="StorageAccount:Key" value="Replace your Storage account key" />

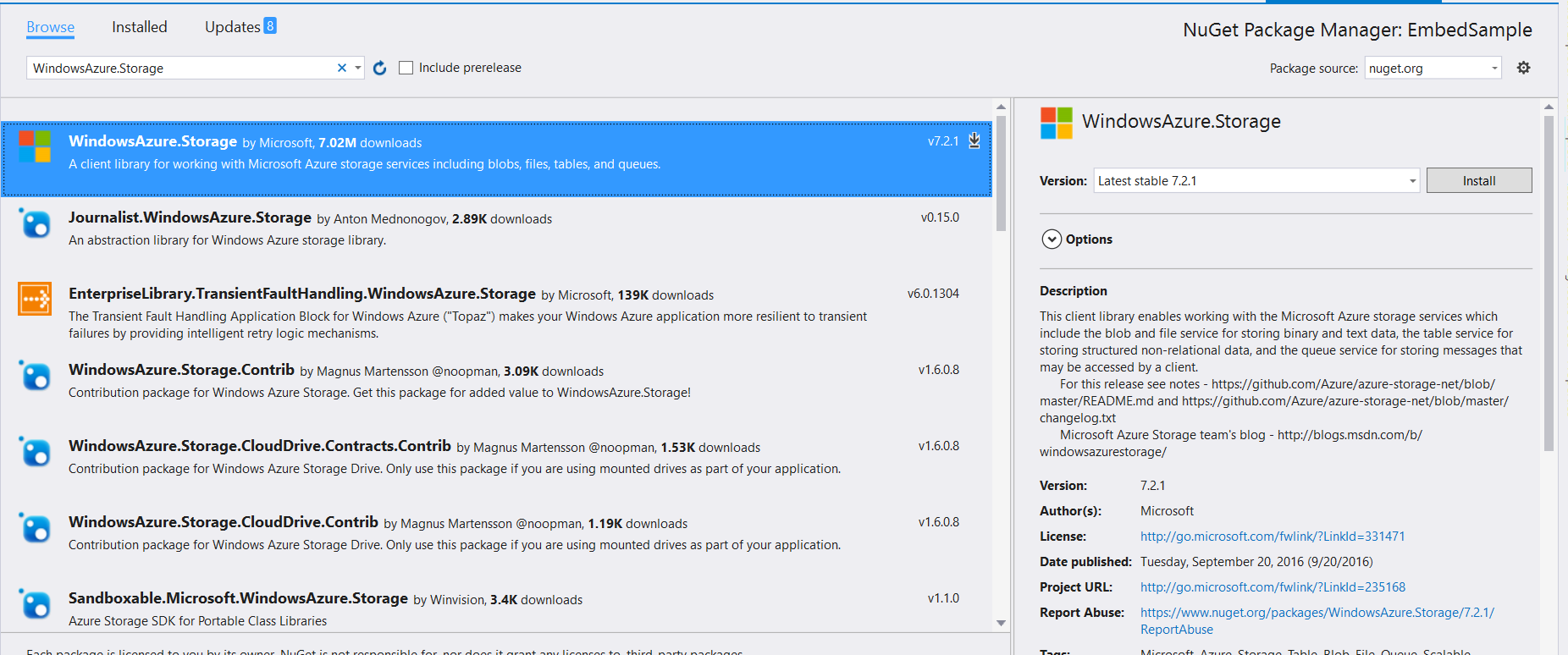
<add key="ServiceBus:ConnectionString" value="Endpoint=sb://[your namespace].servicebus.windows.net;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=[your secret]"/>



* Install the **WindowsAzure.ServiceBus** packagefor Service Bus

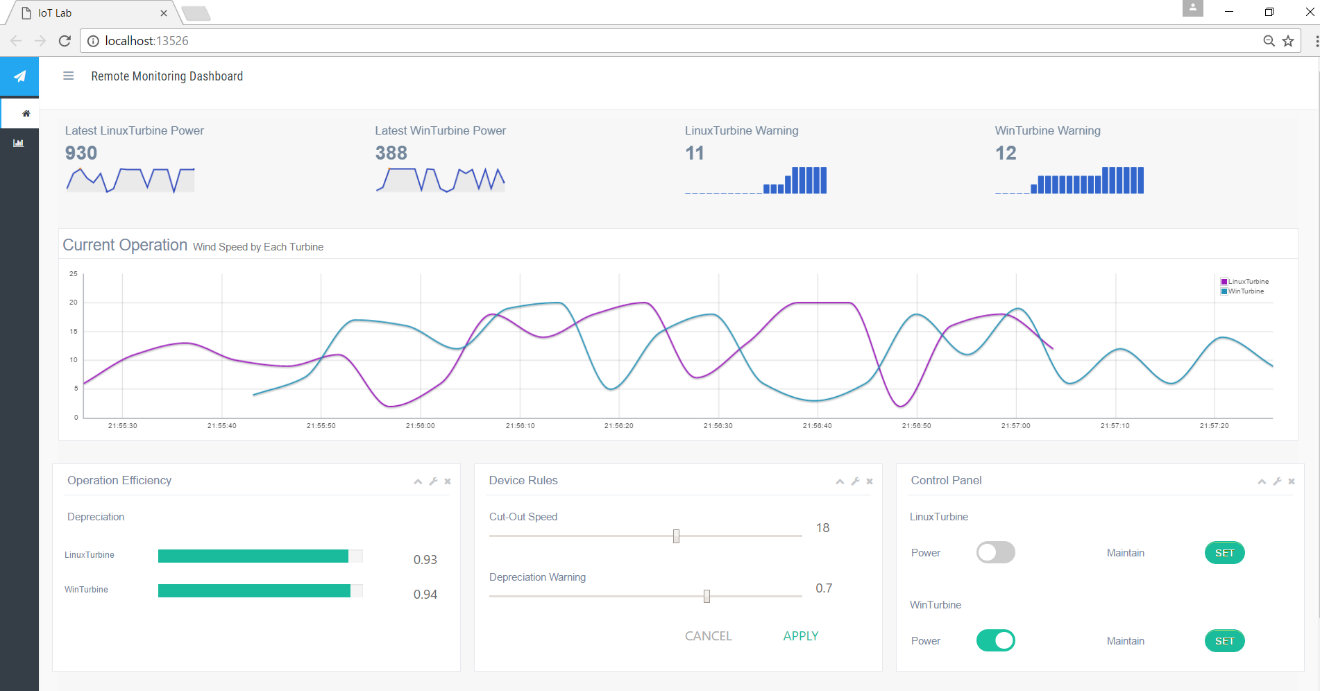


* Install the **WindowsAzure.Storage** packagefor Blob Storage

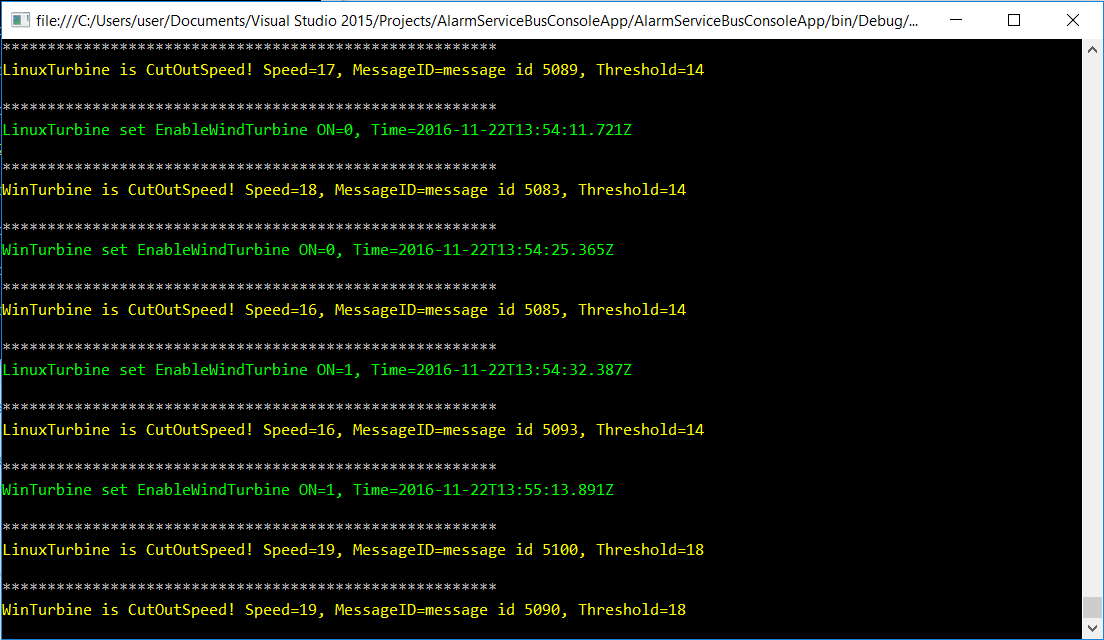


## Step 2: Run all applications

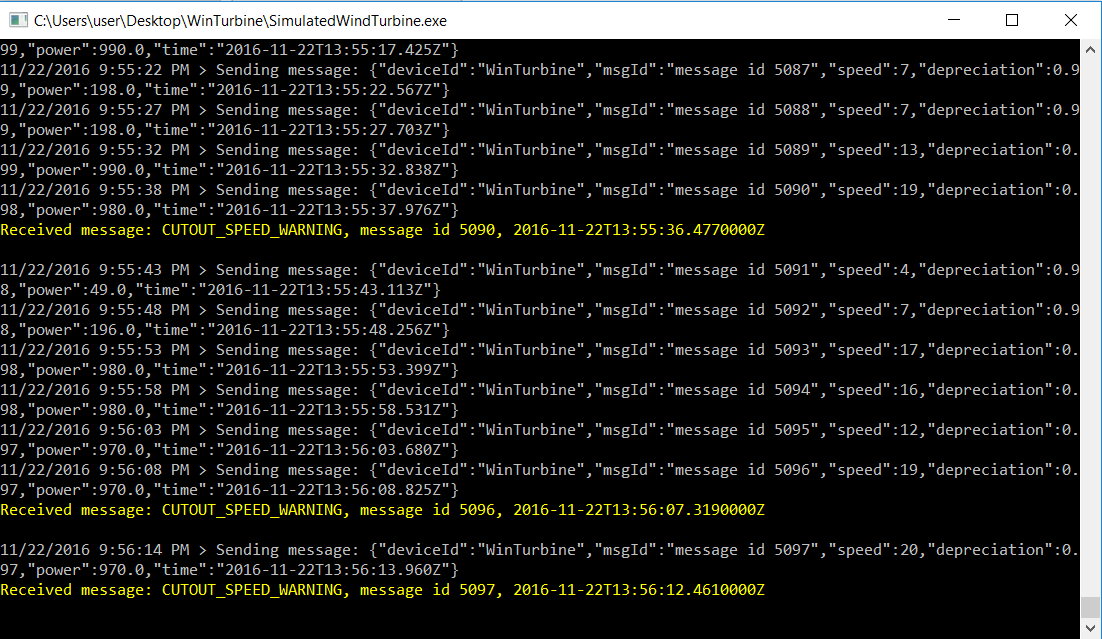
* Run the **Linux** and **Windows** **simulated wind turbines**
* Run the **Alert Service Bus** Console App
* Run the **Telemetry Event Processor Host** Console App
* Run the **Power BI Embedded Web App**, and you can
  + Control the ON/OFF and maintain of device
  + Set the alarm rules of device (will be affected after 2 minutes)



* + Watch the log of Alarm Service Bus



* + Check the C2D log of Device (WinTurbine)



* *The HOL 11 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.*
* *Congratulations! Let’s go the last one thing, to publish the background tasks to Web Job.*