**About this library:**

This is a reusable component for ingesting data to Azure IotHub and EventHub, developed in dot net core (V6.0) and dot net standard (V2.1) and targeted for the Dot Net developer’s community.

The technical documentation and example samples are provided for the developers to quick start with it with minimal time to be spent on using and learning this library.

All technical documentation can be found in the Doc folder and can be accessed by clicking the Index.html page.

**Why this library?**

* Developers with no or little exposure can implement a data ingestion module for Azure IotHub in minutes.
* Efficient and optimized way to send messages to IoTHub with option of sending single messages and batch messages (Most optimize way to send), thus optimized for sending the messages with minimizing the message quota usage of IoTHub tiers.
* No fixed format for the message, thus giving the user/developer the choice to design the json message formats as per the application need.
* Built in properties for better control of the message ingestion and control activities.

Exp.

* Setting the send interval
* To send a single message (with no time gap) or batch messages
* Keeping track of the number of messages sent and bytes sent to hub.
* Failed messages count and last messages sent time etc.
* Resiliency mechanism built-in, thus fail safe. It retries the messages which are failed to be sent to IotHub when there is a network issue. It keeps the failed messages in-memory and tries to send them when the connection is up)
* Supports all IotHub protocols like Mqtt, Http, AMQP etc. for communication.
* Automatically takes care of arranging the messages as 256kb packet in size (when messages set as batch) to efficiently use the IotHub quotas thus avoiding the throttling.

**How to use this library (with 5 simple steps)?**

* **Step 1**

Get the connection string for the IotHub

string iotHubConnString = "HostName=xxx.azure-devices.net;DeviceId=EdgeDevice;SharedAccessKey=xxxxxxxxxxxxxxxxxxx";

* **Step 2**

Create the IotHub Client from the connection string.

IotHubClient iotHubClient = new IotHubClient(iotHubConnString);

* **Step 3**

Create the IotHub Communication module.

IotHubCommunication iotHubCommunication = new IotHubCommunication(iotHubClient);

* **Step 4**

Initialize the IotHub Communication module.

iotHubCommunication.StartHubCommunicationAsync();

* **Step 5**

Start queuing your messages(json) to the internal queue, the component will automatically take care of rest of the things.

iotHubCommunication.EnqueueMessage(jsonPayload);