

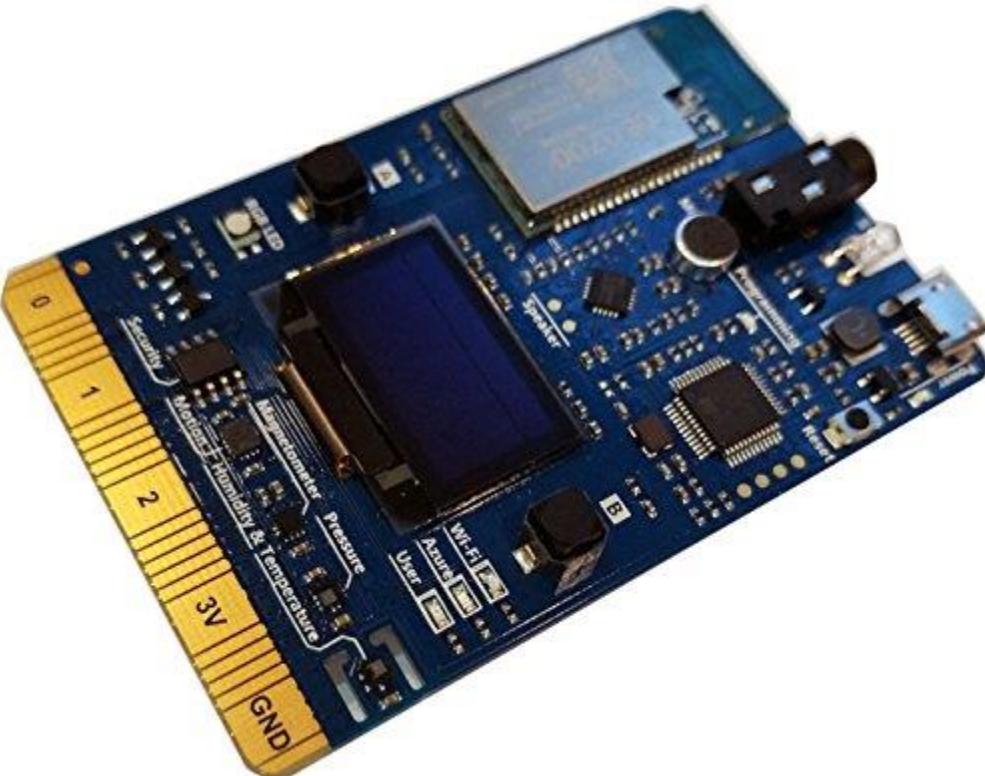
Monitoring IoT appliances in real time

Mark Allibone

Mobile Lead

Rey Automation

@mallibone



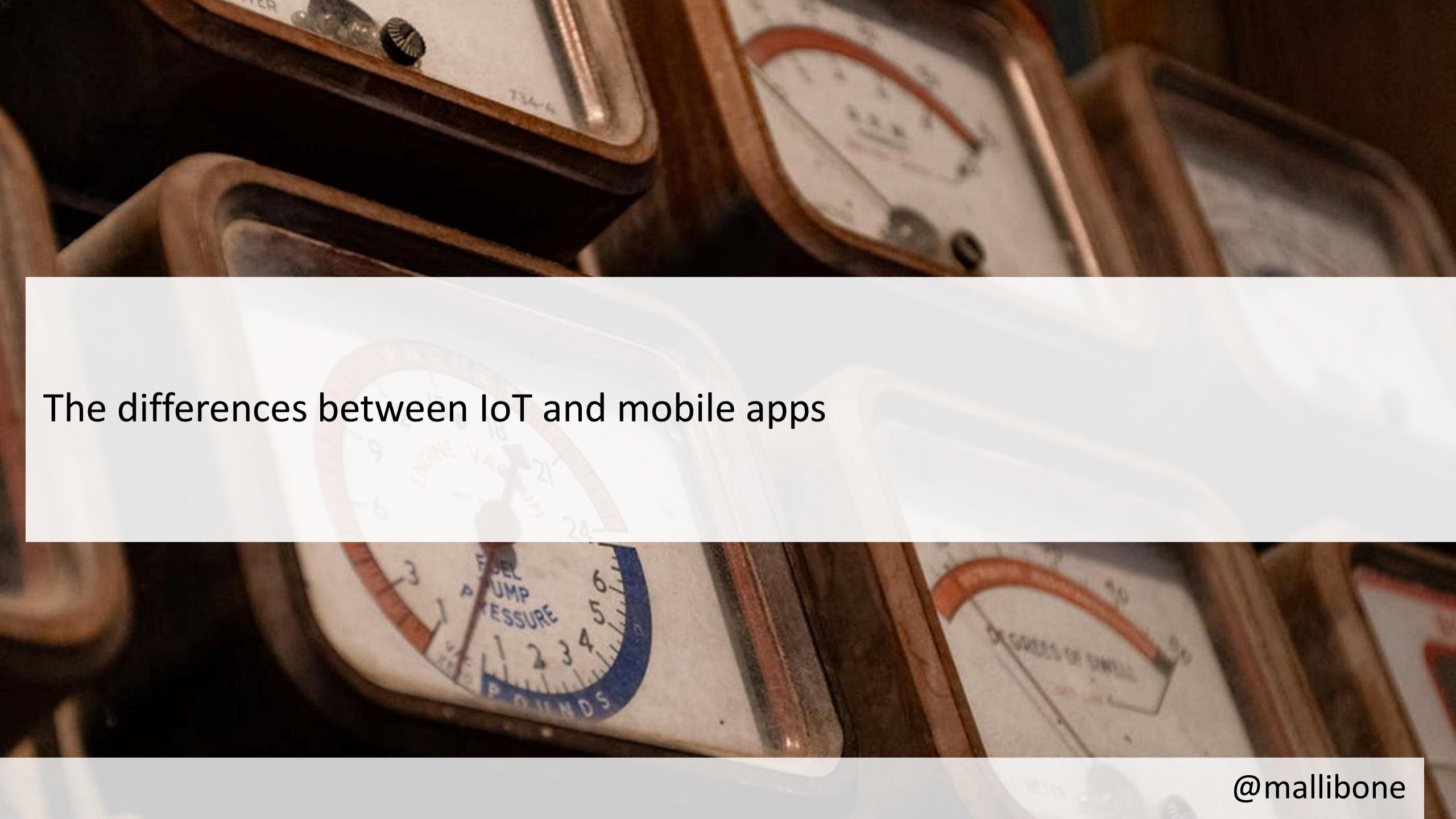
totalgadgetsite.com

@mallibone

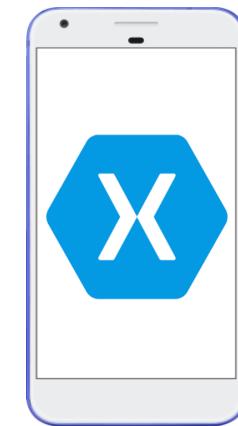
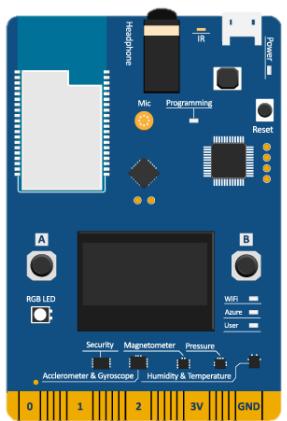


mscloud.be

@mallibone

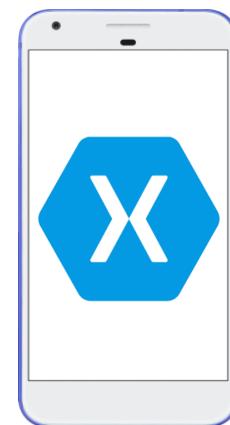
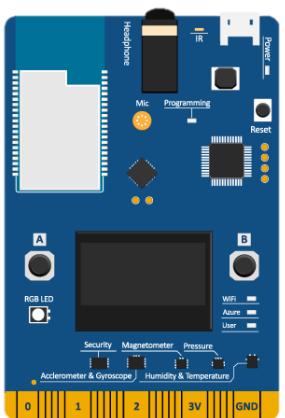


The differences between IoT and mobile apps



@mallibone

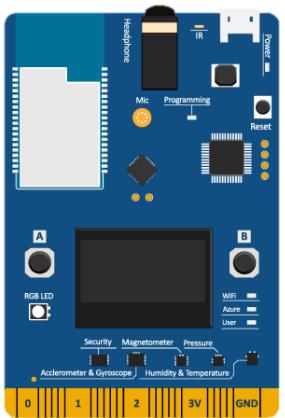
Bluetooth



@mallibone



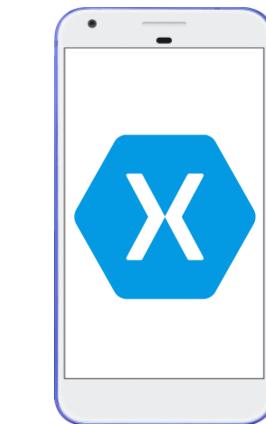
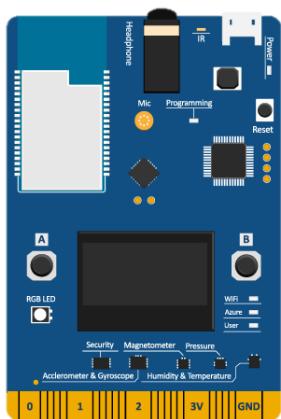
WiFi



@mallibone

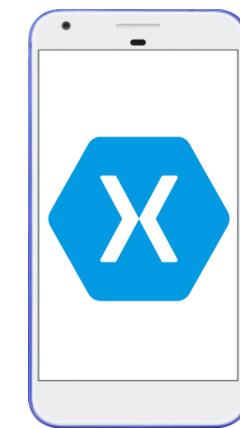
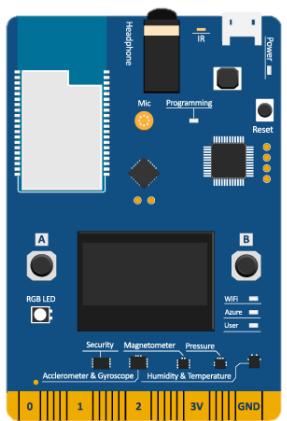
Backend

Probably something involving Azure



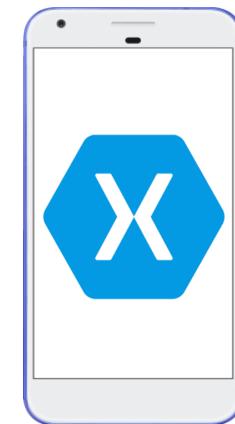
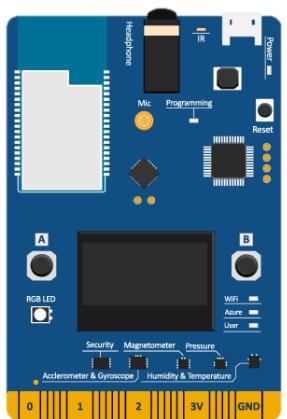


“To the cloud!”



@mallibone

Where the magic
happens



Azure Services Reference Map <https://nnmer.github.io/azure-services-map/dist/>

Azure Services Reference Found an issue or have a suggestion? [Star](#) 72 [Edit](#)

Azure services

Table view [Map view](#)

Search

Show services with In/Out connections only

Note:

- services may be placed in several service groups;
- services may have direct links to how-to connect docs;
- a service with input/output connection

AI + Machine Learning	Analytics	Azure Stack	Compute	Containers	Databases	DevOps	Developer Tools	Identity	Integration	Internet of Things	Management Tools	Media	Migration
Academic Knowledge API	Apache Spark for Azure HDInsight	Azure Stack Operator	App Service	App Service	Azure Cache for Redis	Application Insights	Azure DevOps	Azure Active Directory	API Management	API Management	Application Insights	Azure Media Player	Azure Database Migration Service
Anomaly Finder	Apache Storm for HDInsight	Azure Stack User	Azure Container Instances	Azure Container Instances	Azure Cosmos DB	Azure DevOps	Blockchain Workbench	Azure Active Directory B2C	Event Grid	Azure Cosmos DB	Automation	Content Protection	Azure Migrate
Answer Search	Azure Analysis Services		Azure Kubernetes Service (AKS)	Azure Dev Spaces	Azure Data Explorer	Azure DevOps Projects	CLI	Azure Active Directory for Domain Services	Logic Apps	Azure Digital Twins	Azure Advisor	Encoding	Cost Management
Azure Bot Service	Azure Data Explorer		Batch	Azure Kubernetes Service (AKS)	Azure Database for MariaDB	Azure Lab Services	Developer Tool Integrations	Azure Information Protection	Service Bus	Azure IoT	Azure Blueprints	Live and On-Demand Streaming	Site Recovery
Azure Databricks	Azure Databricks		Cloud Services	Batch	Azure Database for MySQL	Developer Tool Integrations	SDKs	Multi-Factor Authentication		Azure Maps	Azure Managed Applications	Media Analytics	
Azure Notebooks	Data Catalog		CycleCloud	Container Registry	Azure Database for PostgreSQL	HockeyApp	Visual Studio	Security Information		Azure Sphere	Azure Migrate	Media Services	
Azure Search	Data Factory		Functions	Service Fabric	Azure Database Migration Service	Visual Studio App Center	Visual Studio Code			Event Grid	Azure mobile app	Video Indexer	
Bing Autosuggest API	Data Lake Analytics		Linux Virtual Machines	Web App for Containers	Azure SQL Database					Functions	Azure Monitor		
Bing Custom Search API	Data Lake Storage Gen2		SAP HANA on Azure Large Instances		Data Factory					IoT Central	Azure Policy		

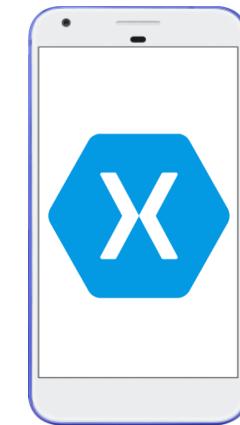
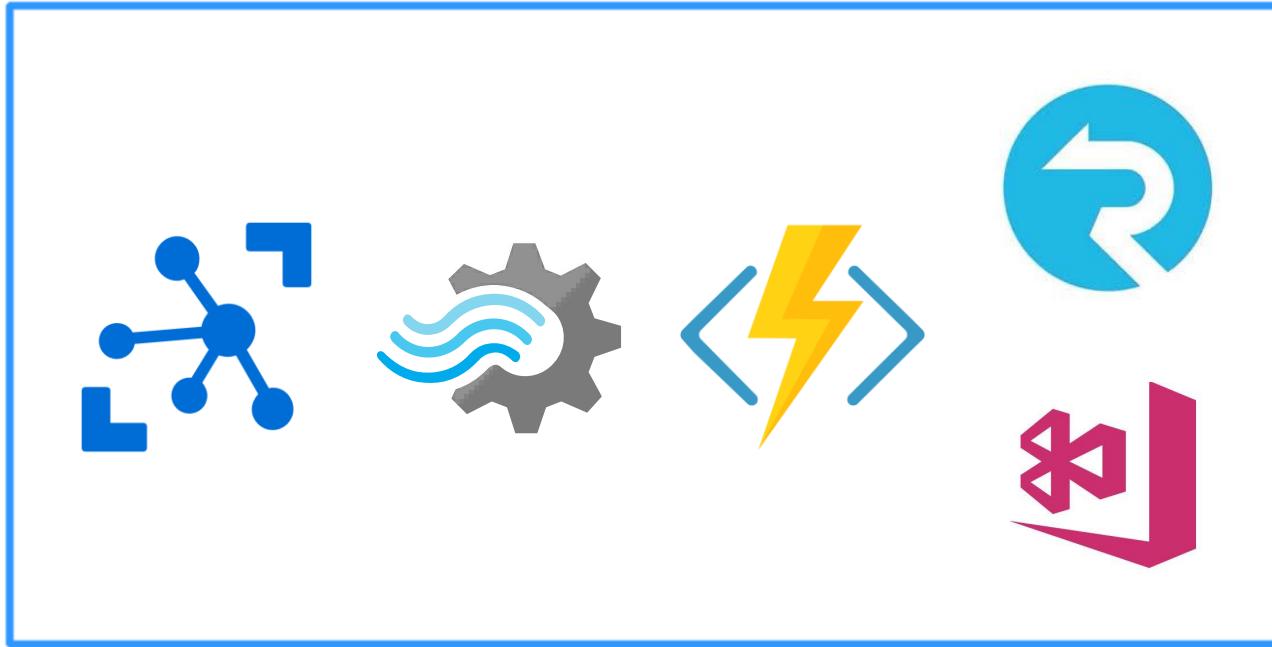
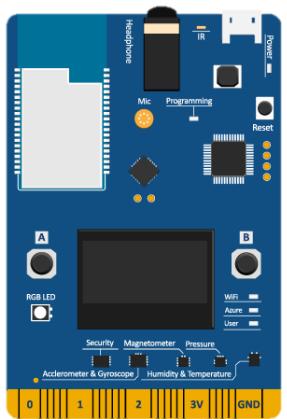


<https://nnmer.github.io/azure-services-map/dist/>

@mallibone

Azure Resources 101

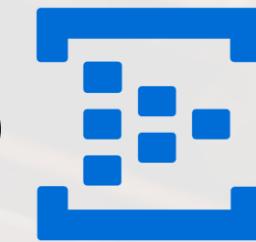
- Group your resources
- Create services in the same region/datacenter
- Check Pricing / Limitations



@mallibone

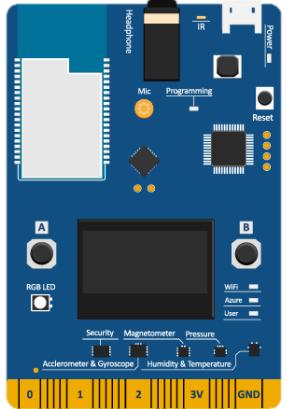


IoT Hub or Event Hub



IoT Hub or Event Hub?

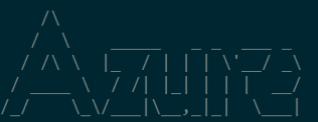
IoT Hub	Event Hub
Communication from Device to Cloud and Cloud to Device	Communication from Client to Cloud
Millions of Concurrent Connections (AMQP)	5000 Concurrent Connections (AMQP)
Integrated Device Management	No Device Management



Azure Cloud

@mallibone

C:\Work\Talks> az



Welcome to the cool new Azure CLI!

Use `az --version` to display the current version.

Here are the base commands:

account	: Manage Azure subscription information.
acr	: Manage private registries with Azure Container Registries.
acs	: Manage Azure Container Services.
ad	: Manage Azure Active Directory Graph entities needed for Role Based Access Control.
advisor	: Manage Azure Advisor.

aks	: Manage Azure Kubernetes Services.
ams	: Manage Azure Media Services resources.

appservice	: Manage App Service plans.
------------	-----------------------------

backup	: Manage Azure Backups.
--------	-------------------------

batch	: Manage Azure Batch.
-------	-----------------------

batchai	: Manage Batch AI resources.
---------	------------------------------

billing	: Manage Azure Billing.
---------	-------------------------

bot	: Manage Microsoft Bot Services.
-----	----------------------------------

cognitiveservices	: Manage Azure Cognitive Services accounts.
-------------------	---

configure	: Manage Azure CLI configuration. This command is interactive.
-----------	--

core	: Manage Azure CLI core resources.
------	------------------------------------

cosmosdb	: Manage Azure Cosmos DB database accounts.
----------	---

deployment	: Manage Azure Resource Manager deployments at subscription scope.
------------	--

disk	: Manage Azure Managed Disks.
------	-------------------------------

dla	: (PREVIEW) Manage Data Lake Analytics accounts, jobs, and catalogs.
-----	--

dls	: (PREVIEW) Manage Data Lake Store accounts and filesystems.
-----	--

dms	: Manage Azure Data Migration Service (DMS) instances.
-----	--

eventgrid	: Manage Azure Event Grid topics and subscriptions.
-----------	---

eventhubs	: Manage Azure Event Hubs namespaces, eventhubs, consumergroups and geo recovery configurations - Alias.
-----------	--

extension	: Manage and update CLI extensions.
-----------	-------------------------------------

feature	: Manage resource provider features.
---------	--------------------------------------

feedback	: Send feedback to the Azure CLI Team!
----------	--

find	: I'm an AI robot, my advice is based on our Azure documentation as well as the usage patterns of Azure CLI and Azure ARM users. Using me improves Azure products and documentation.
------	--

functionapp	: Manage function apps.
-------------	-------------------------

group	: Manage resource groups and template deployments.
-------	--

hdinsight	: Manage HDInsight resources.
-----------	-------------------------------

identity	: Managed Service Identities.
----------	-------------------------------

image	: Manage custom virtual machine images.
-------	---

interactive	: Start interactive mode. Installs the Interactive extension if not installed already.
-------------	--

iot	: Manage Internet of Things (IoT) assets. Augmented with the IoT extension.
-----	---

iotcentral	: Manage IoT Central assets.
------------	------------------------------

keyvault	: Manage KeyVault keys, secrets, and certificates.
----------	--

kusto	: Manage Azure Kusto resources.
-------	---------------------------------

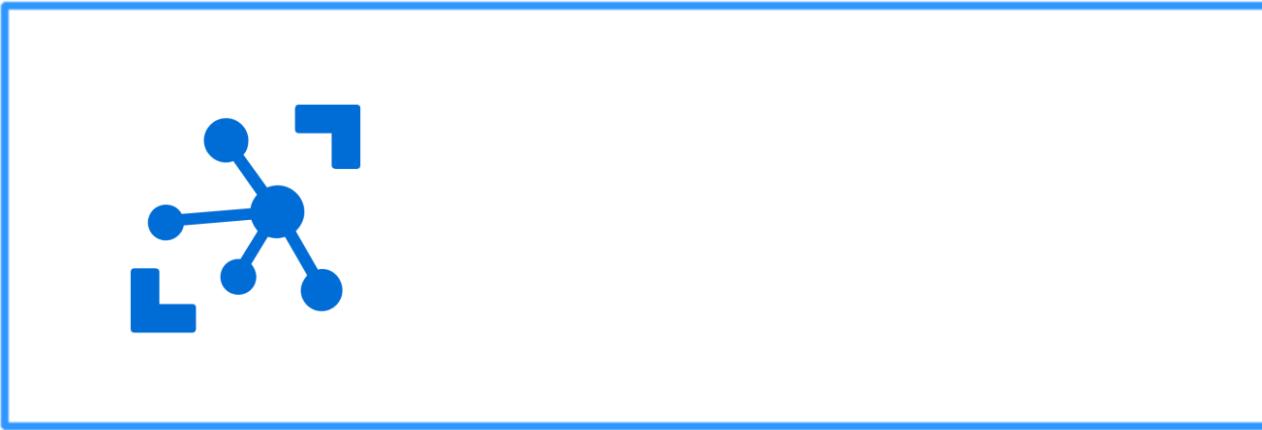
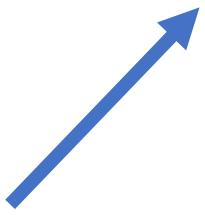
lab	: Manage Azure DevTest Labs.
-----	------------------------------

lock	: Manage Azure locks.
------	-----------------------

login	: Log in to Azure.
-------	--------------------

<https://github.com/Azure/azure-iot-cli-extension#installation>

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli?view=azure-cli-latest>



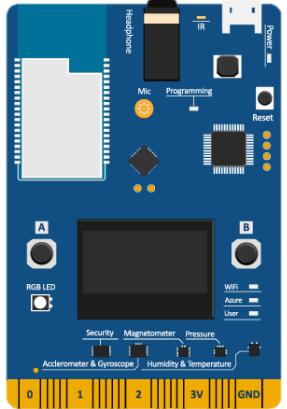
Azure Cloud

DEMO

@mallibone



@mallibone



Azure Cloud

@mallibone

IoT DevKit to cloud -- Connect to cloud

https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-arduino-iot-devkit-az3166-get-started

NoserEngineering

Microsoft Azure

Overview Solutions Products Documentation Pricing Training Marketplace Partners Support Blog More Free account >

Azure / Internet of Things / IoT Hub

Learn Azure at your own pace See training modules >

Filter by title

IoT Hub Documentation

- > Overview
- > Quickstarts
- > Tutorials
- > Concepts
- > How-to guides
 - > Develop
 - > Manage
 - > Use real devices
 - Use an online simulator
 - > Use a physical device
 - Raspberry Pi with Node.js
 - Raspberry Pi with C
- MXChip IoT DevKit with Arduino**
- Adafruit Feather HUZZAH ESP8266 with Arduino
 - > Use MXChip IoT DevKit
 - > Extended IoT scenarios
 - > Troubleshoot
 - > Reference
 - > Related
 - > Resources

04/17/2019 • 12 minutes to read • Contributors all

IoT DevKit AZ3166 with VS Code

You can use the [MXChip IoT DevKit](#) to develop and prototype Internet of Things (IoT) solutions that take advantage of Microsoft Azure services. It includes an Arduino-compatible board with rich peripherals and sensors, an open-source board package, and a rich [sample gallery](#).

What you learn

- How to create an IoT hub and register a device for the MXChip IoT DevKit.
- How to connect the IoT DevKit to Wi-Fi and configure the IoT Hub connection string.
- How to send the DevKit sensor telemetry data to your IoT hub.
- How to prepare the development environment and develop application for the IoT DevKit.

Don't have a DevKit yet? Try the [DevKit simulator](#) or [purchase a DevKit](#).

What you need

- A MXChip IoT DevKit board with a Micro-USB cable. [Get it now](#).
- A computer running Windows 10, macOS 10.10+ or Ubuntu 18.04+.
- An active Azure subscription. [Activate a free 30-day trial Microsoft Azure account](#).

Open Azure Cloud Shell

Azure Cloud Shell is a free, interactive shell that you can use to run the steps in this article. Common Azure tools are preinstalled and configured in Cloud Shell for you to use with your account. Select **Copy** to copy the code, paste it in Cloud Shell, and then press Enter to run it. There are a few ways to open Cloud Shell:

Select Try It in the upper-right corner of a code block.

Azure CLI

Open Cloud Shell in your browser.

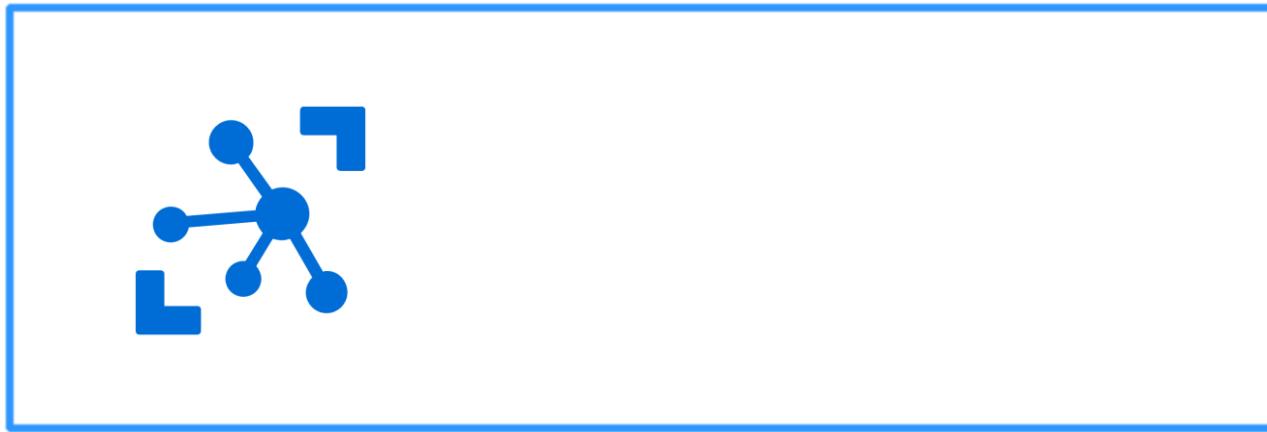
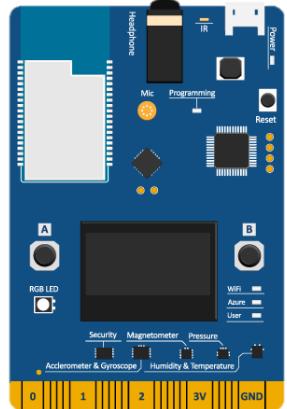
Launch Cloud Shell

Select the Cloud Shell button on the menu in the upper-right corner of the Azure portal.

Is this page helpful? Yes No

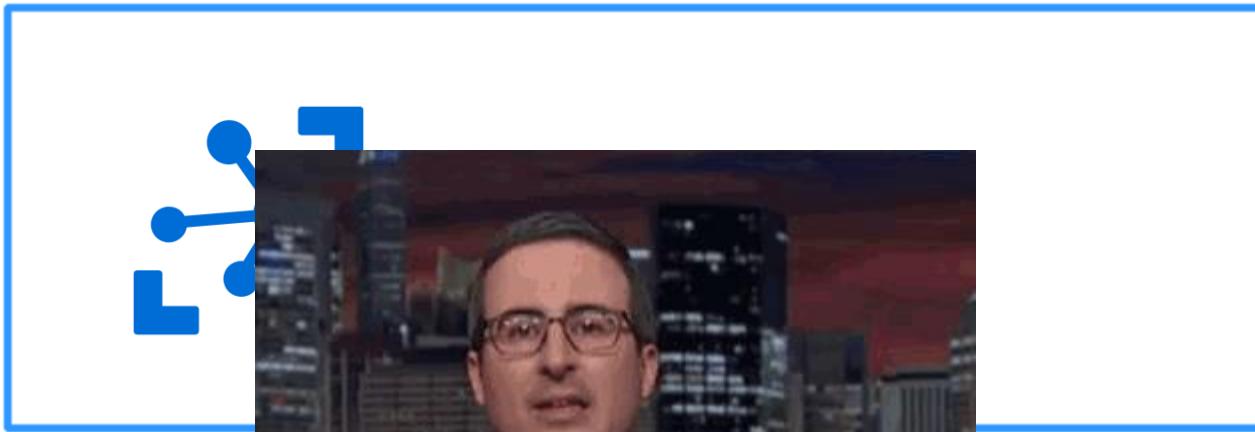
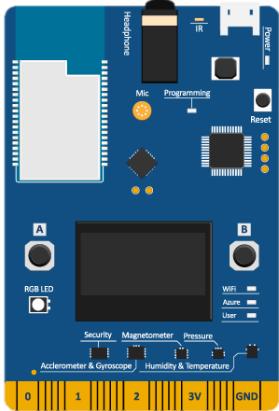
<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-arduino-iot-devkit-az3166-get-started>

@mallbone

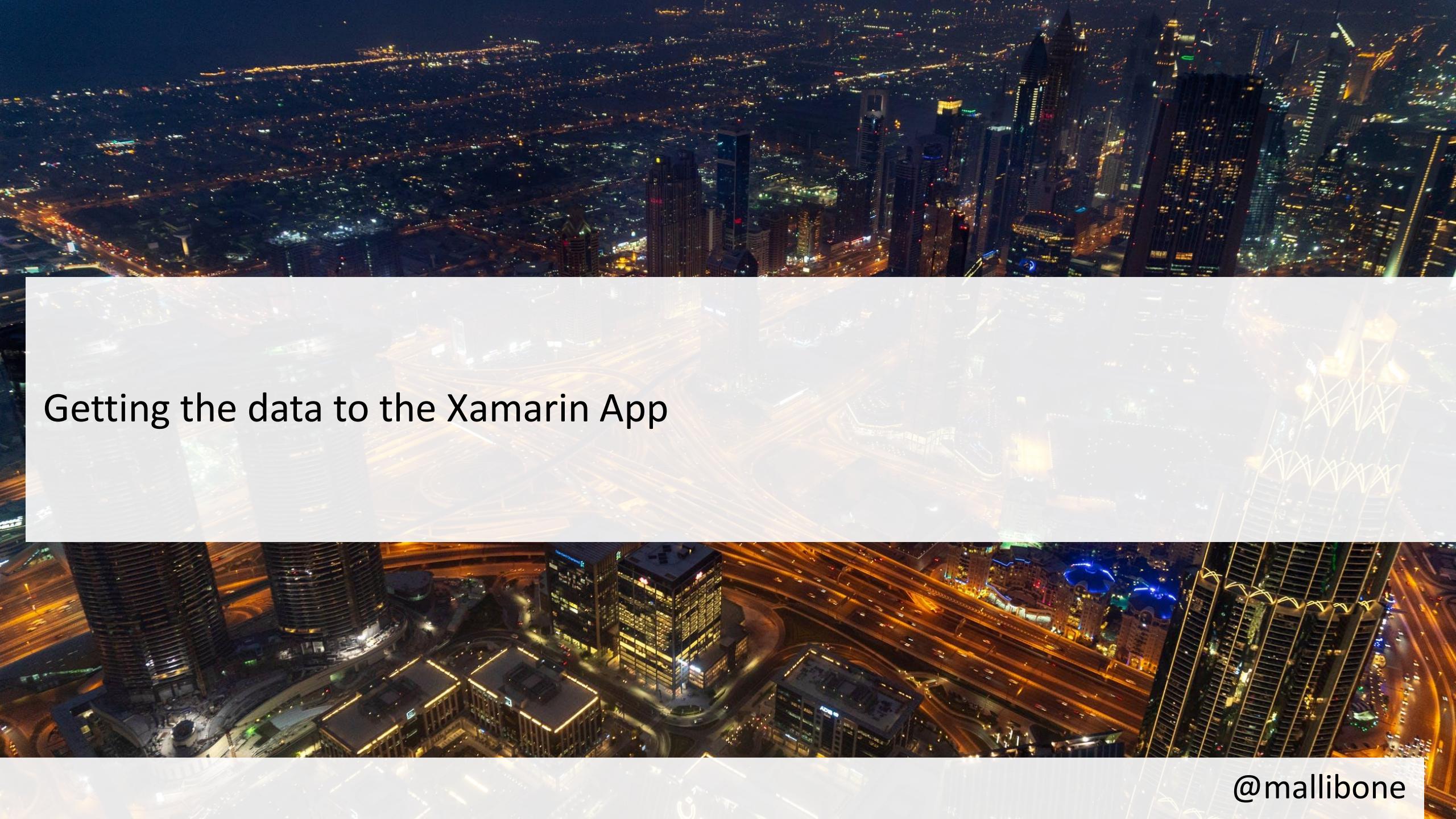


Azure Cloud

@mallibone



@mallibone



Getting the data to the Xamarin App

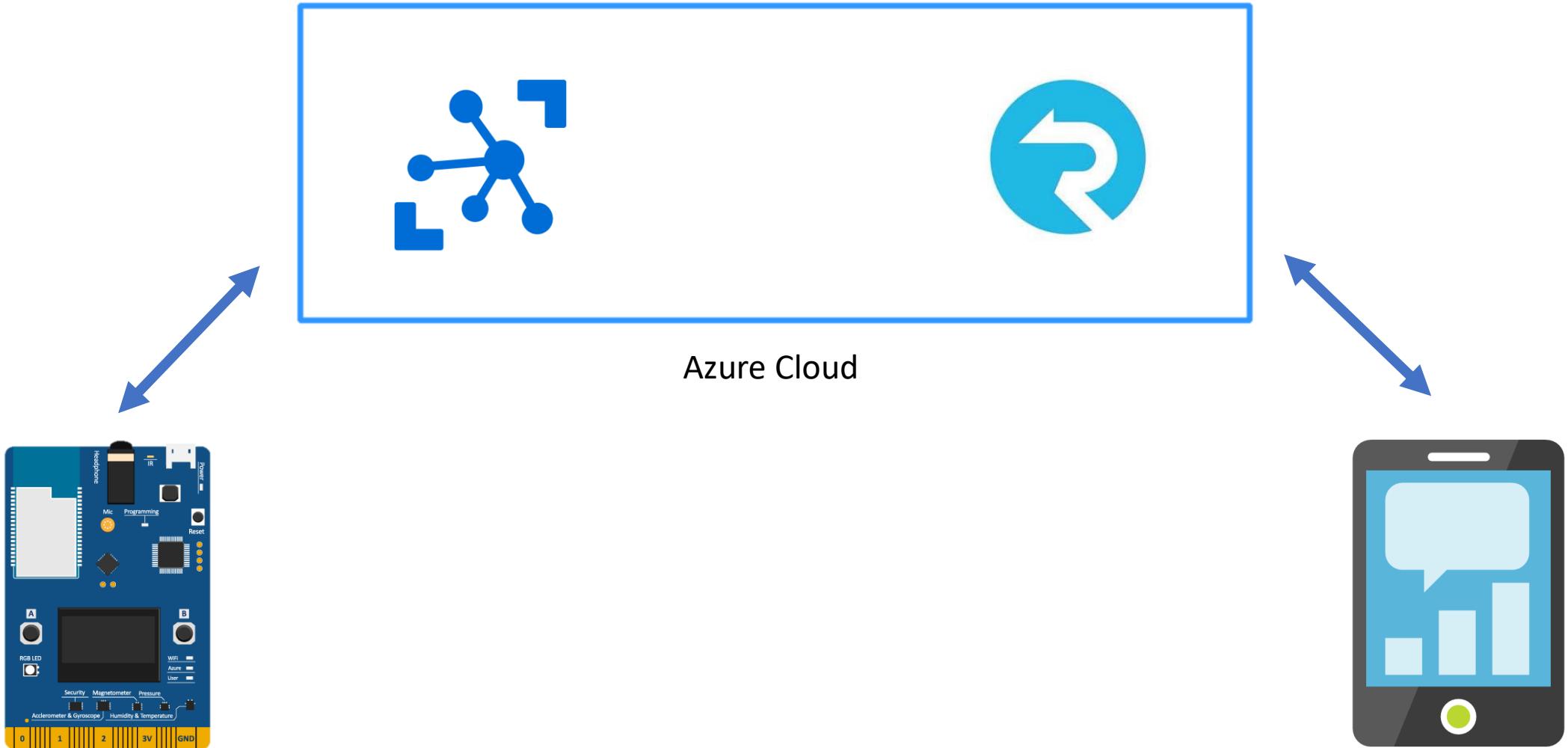
@mallibone



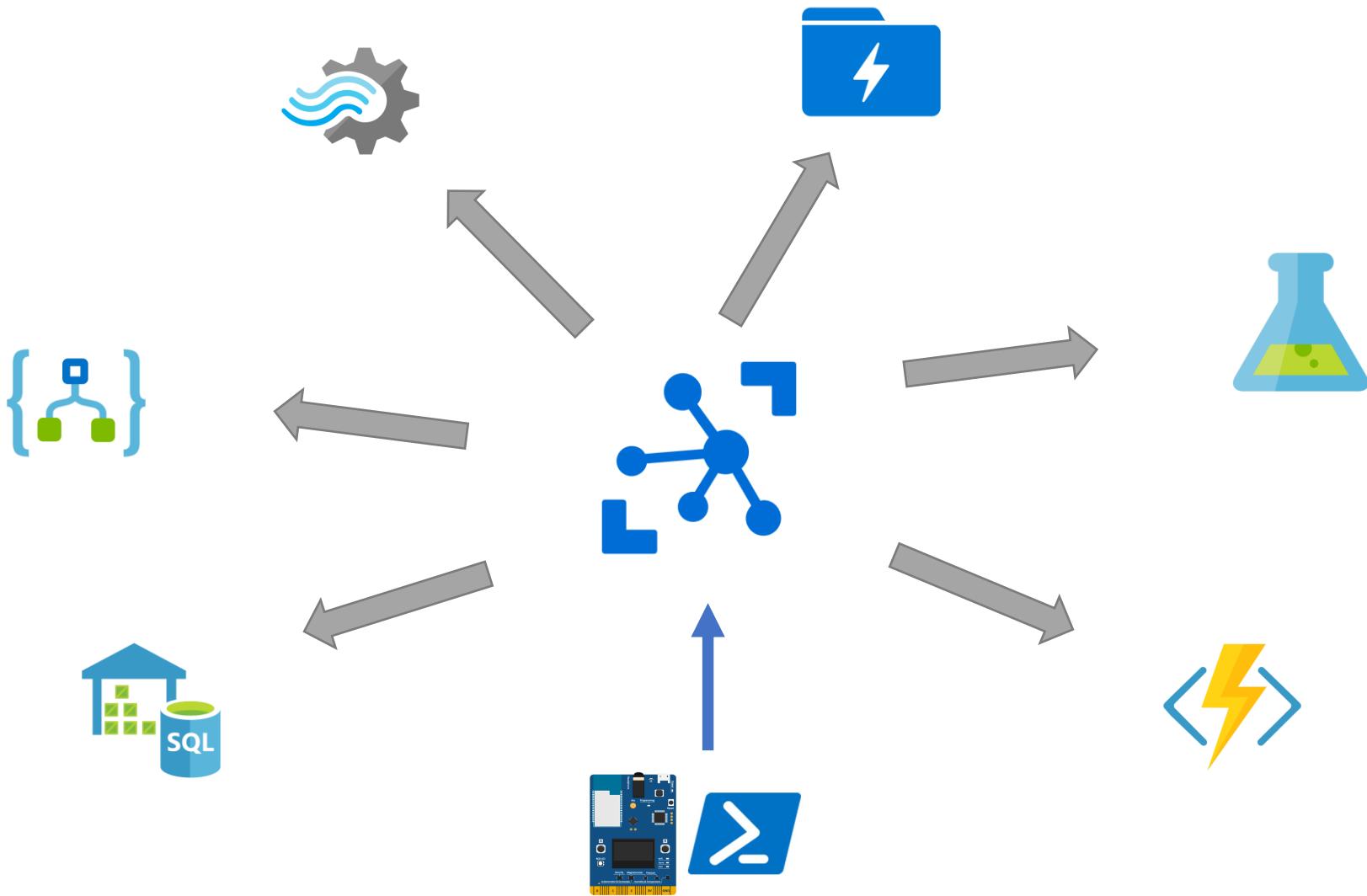
Poll vs Push



@mallibone



@mallibone



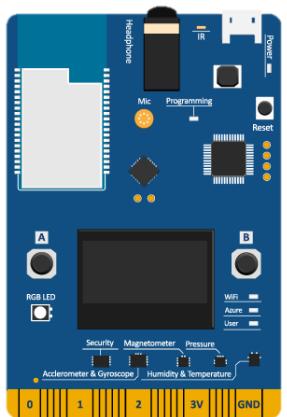
@mallibone



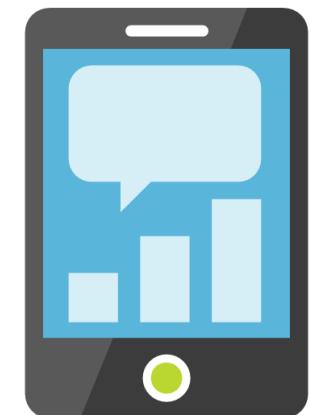
Processing IoT Hub data



@mallibone



Azure Cloud



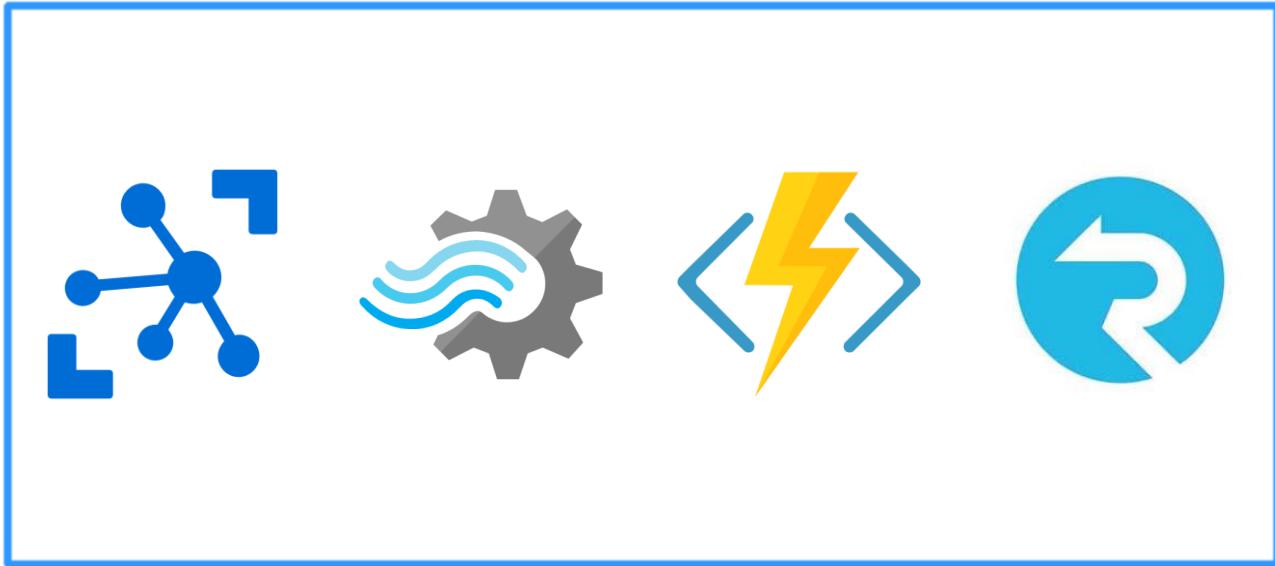
@mallibone



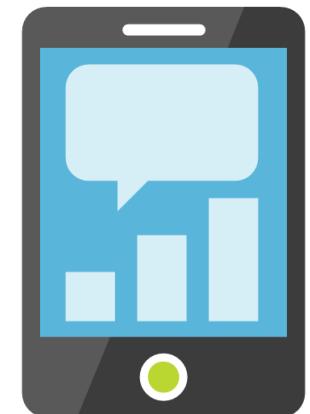
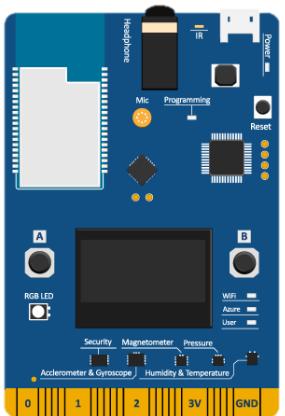
Azure Functions



@mallibone



Azure Cloud



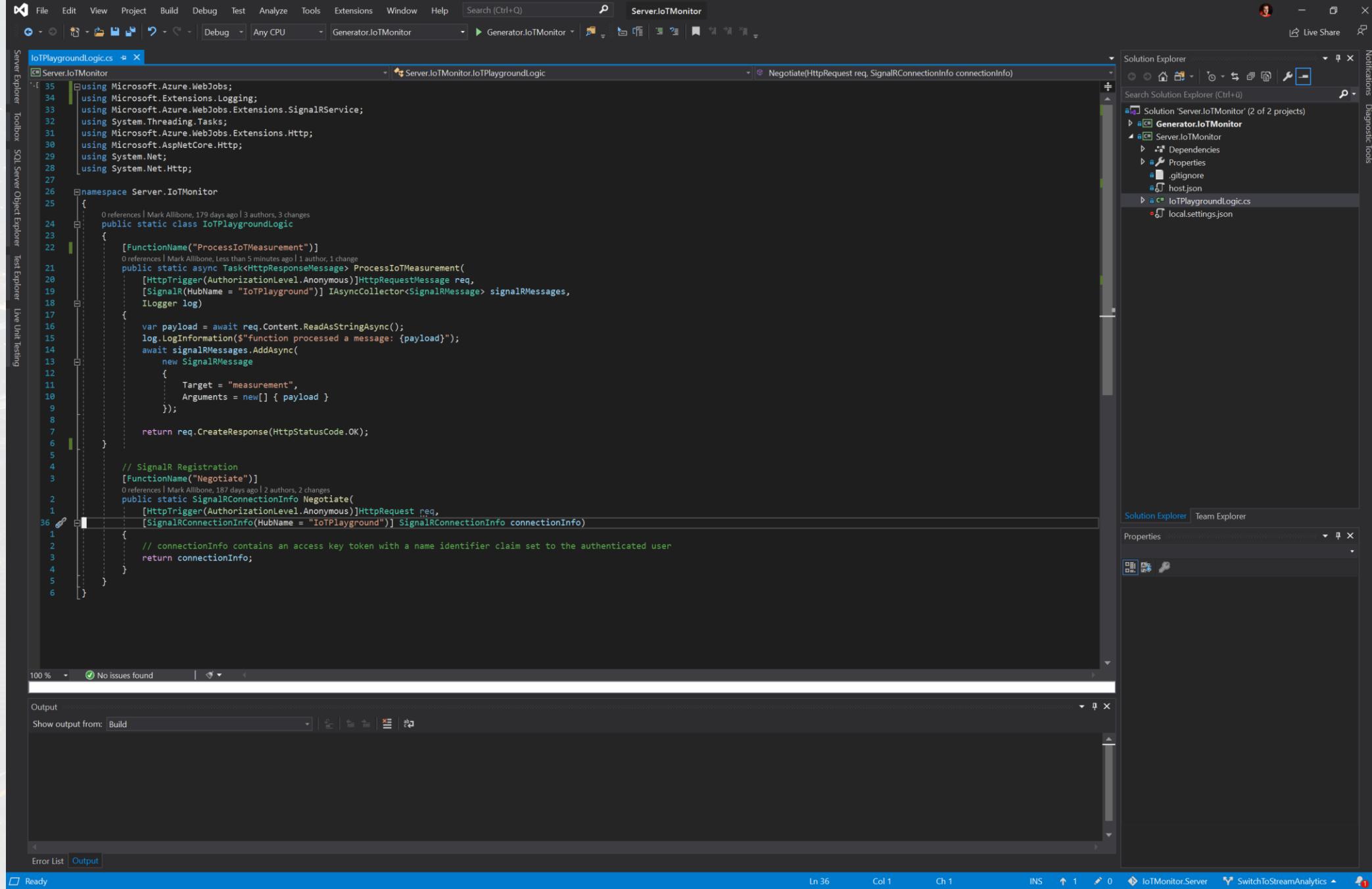
@mallibone



SignalR



@mallibone



@mallibone

```
var payload = await req.Content.ReadAsStringAsync();
log.LogInformation($"function processed a message: {payload}");
await signalRMessages.AddAsync(
    new SignalRMessage
    {
        Target = "measurement",
        Arguments = new[] { payload }
    });
}

return req.CreateResponse(HttpStatusCode.OK);
}

// SignalR Registration
[FunctionName("Negotiate")]
0 references | Mark Allibone, 187 days ago | 2 authors | 2 changes
public static SignalRConnectionInfo Negotiate(
    [HttpTrigger(AuthorizationLevel.Anonymous)] HttpRequest req,
    [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
{
    // connectionInfo contains an access key token with a name identifier claim set to the authenticated user
    return connectionInfo;
}
```

No issues found

@mallibone

```
var payload = await req.Content.ReadAsStringAsync();
log.LogInformation($"function processed a message: {payload}");
await signalRMessages.AddAsync(
    new SignalRMessage
    {
        Target = "measurement",
        Arguments = new[] { payload }
    });
}

return req.CreateResponse(HttpStatusCode.OK);
}

// SignalR Registration
[FunctionName("Negotiate")]
public static SignalRConnectionInfo Negotiate(
    [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequest req,
    [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
{
    // connectionInfo contains an access key token with a name identifier claim set to the authenticated user
    return connectionInfo;
}
```

```
var payload = await req.Content.ReadAsStringAsync();
log.LogInformation($"function processed a message: {payload}");
await signalRMessages.AddAsync(
    new SignalRMessage
    {
        Target = "measurement",
        Arguments = new[] { payload }
    });
}

return req.CreateResponse(HttpStatusCode.OK);
}

// SignalR Registration
[FunctionName("Negotiate")]
0 references | Mark Allibone, 187 days ago | 2 authors, 2 changes
public static SignalRConnectionInfo Negotiate(
    [HttpTrigger(AuthorizationLevel.Anonymous)] HttpRequest req,
    [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
{
    // connectionInfo contains an access key token with a name identifier claim set to the authenticated user
    return connectionInfo;
}
```

```
var payload = await req.Content.ReadAsStringAsync();
log.LogInformation($"function processed a message: {payload}");
await signalRMessages.AddAsync(
    new SignalRMessage
    {
        Target = "measurement",
        Arguments = new[] { payload }
    });
}

return req.CreateResponse(HttpStatusCode.OK);
}

// SignalR Registration
[FunctionName("Negotiate")]
0 references | Mark Allibone, 187 days ago | 2 authors, 2 changes
public static SignalRConnectionInfo Negotiate(
    [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequest req,
    [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
{
    // connectionInfo contains an access key token with a name identifier claim set to the authenticated user
    return connectionInfo;
}
```

No issues found

@mallibone

```
var payload = await req.Content.ReadAsStringAsync();
log.LogInformation($"function processed a message: {payload}");
await signalRMessages.AddAsync(
    new SignalRMessage
    {
        Target = "measurement",
        Arguments = new[] { payload }
    });
}

return req.CreateResponse(HttpStatusCode.OK);
}

// SignalR Registration
[FunctionName("Negotiate")]
0 references | Mark Allibone, 187 days ago | 2 authors, 2 changes
public static SignalRConnectionInfo Negotiate(
    [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequest req,
    [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
{
    // connectionInfo contains an access key token with a name identifier claim set to the authenticated user
    return connectionInfo;
}
```

No issues found

@mallibone

```
var payload = await req.Content.ReadAsStringAsync();
log.LogInformation($"function processed a message: {payload}");
await signalRMessages.AddAsync(
    new SignalRMessage
    {
        Target = "measurement",
        Arguments = new[] { payload }
    });
}

return req.CreateResponse(HttpStatusCode.OK);
}

// SignalR Registration
[FunctionName("Negotiate")]
0 references | Mark Allibone, 187 days ago | 2 authors, 2 changes
public static SignalRConnectionInfo Negotiate(
    [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequest req,
    [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
{
    // connectionInfo contains an access key token with a name identifier claim set to the authenticated user
    return connectionInfo;
}
```

```
namespace Server.IoTMonitor
{
    0 references | Mark Allibone, 179 days ago | 3 authors, 3 changes
    public static class IoTPlaygroundLogic
    {
        [FunctionName("ProcessIoTMeasurement")]
        0 references | Mark Allibone, Less than 5 minutes ago | 1 author, 1 change
        public static async Task<HttpResponseMessage> ProcessIoTMeasurement(
            [HttpTrigger(AuthorizationLevel.Anonymous)] HttpRequestMessage req,
            [SignalR(HubName = "IoTPlayground")] IAsyncCollector<SignalRMessage> signalRMessages,
            ILogger log)
        {
            var payload = await req.Content.ReadAsStringAsync();
            log.LogInformation($"function processed a message: {payload}");
            await signalRMessages.AddAsync(
                new SignalRMessage
                {
                    Target = "measurement",
                    Arguments = new[] { payload }
                });
            return req.CreateResponse(HttpStatusCode.OK);
        }

        // SignalR Registration
        [FunctionName("Negotiate")]
        0 references | Mark Allibone, 187 days ago | 2 authors, 2 changes
        public static SignalRConnectionInfo Negotiate(
            [HttpTrigger(AuthorizationLevel.Anonymous)] HttpRequest req,
            [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
        {
    }
```

```
namespace Server.IoTMonitor
{
    0 references | Mark Allibone, 179 days ago | 3 authors, 3 changes
    public static class IoTPlaygroundLogic
    {
        [FunctionName("ProcessIoTMeasurement")]
        0 references | Mark Allibone, Less than 5 minutes ago | 1 author, 1 change
        public static async Task<HttpResponseMessage> ProcessIoTMeasurement(
            [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequestMessage req,
            [SignalR(HubName = "IoTPlayground")] IAsyncCollector<SignalRMessage> signalRMessages,
            ILogger log)
        {
            var payload = await req.Content.ReadAsStringAsync();
            log.LogInformation($"Function processed a message: {payload}");
            await signalRMessages.AddAsync(
                new SignalRMessage
                {
                    Target = "measurement",
                    Arguments = new[] { payload }
                });
        }

        return req.CreateResponse(HttpStatusCode.OK);
    }

    // SignalR Registration
    [FunctionName("Negotiate")]
    0 references | Mark Allibone, 187 days ago | 2 authors, 2 changes
    public static SignalRConnectionInfo Negotiate(
        [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequest req,
        [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
    {
    }
}
```

```
namespace Server.IoTMonitor
{
    0 references | Mark Allibone, 179 days ago | 3 authors, 3 changes
    public static class IoTPlaygroundLogic
    {
        [FunctionName("ProcessIoTMeasurement")]
        0 references | Mark Allibone, Less than 5 minutes ago | 1 author, 1 change
        public static async Task<HttpResponseMessage> ProcessIoTMeasurement(
            [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequestMessage req,
            [SignalR(HubName = "IoTPlayground")] IAsyncCollector<SignalRMessage> signalRMessages,
            ILogger log)
        {
            var payload = await req.Content.ReadAsStringAsync();
            log.LogInformation($"function processed a message: {payload}");
            await signalRMessages.AddAsync(
                new SignalRMessage
                {
                    Target = "measurement",
                    Arguments = new[] { payload }
                });
        }

        return req.CreateResponse(HttpStatusCode.OK);
    }

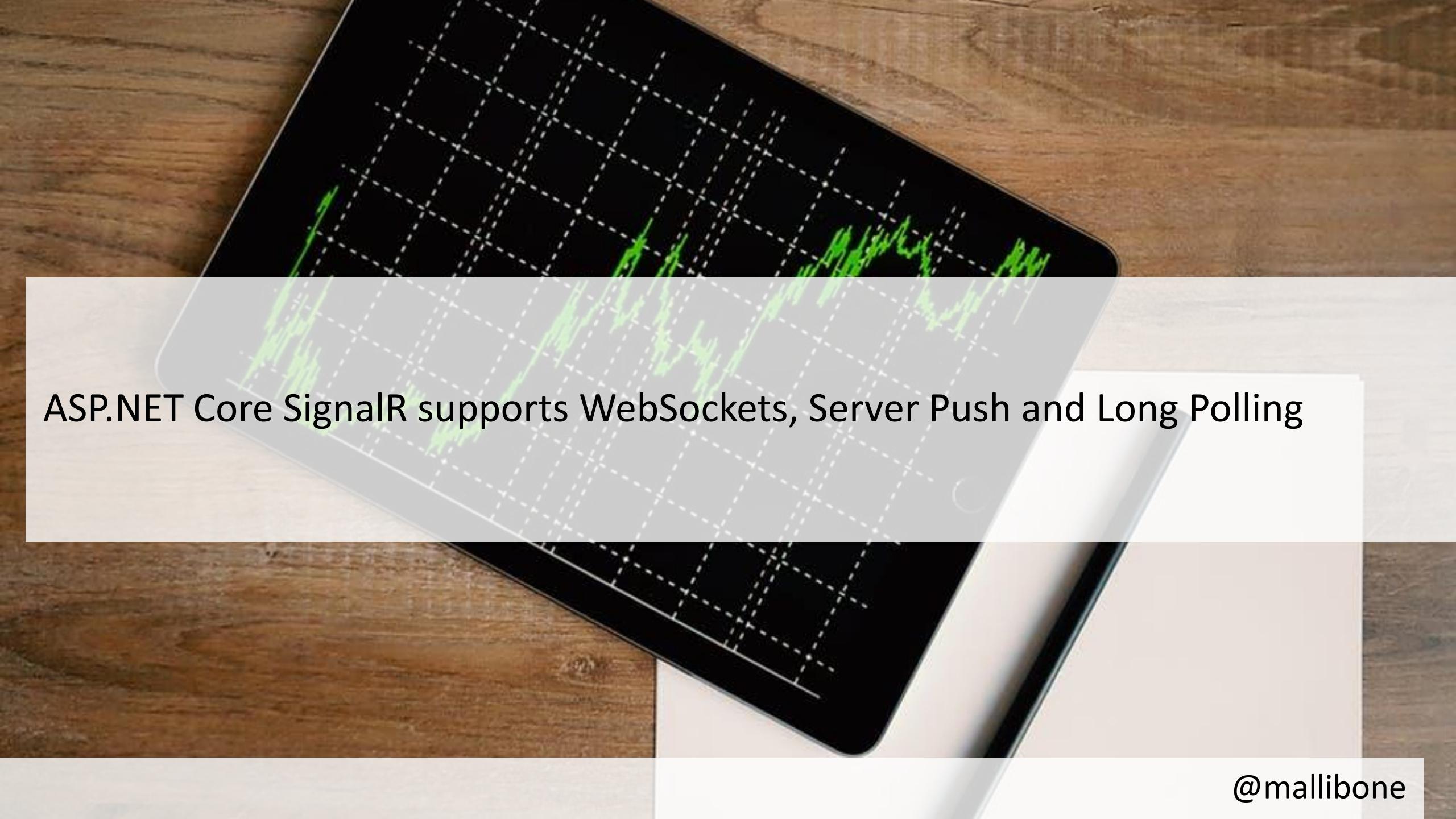
    // SignalR Registration
    [FunctionName("Negotiate")]
    0 references | Mark Allibone, 187 days ago | 2 authors, 2 changes
    public static SignalRConnectionInfo Negotiate(
        [HttpTrigger(AuthorizationLevel.Anonymous)]HttpRequest req,
        [SignalRConnectionInfo(HubName = "IoTPlayground")] SignalRConnectionInfo connectionInfo)
    {
    }
}
```



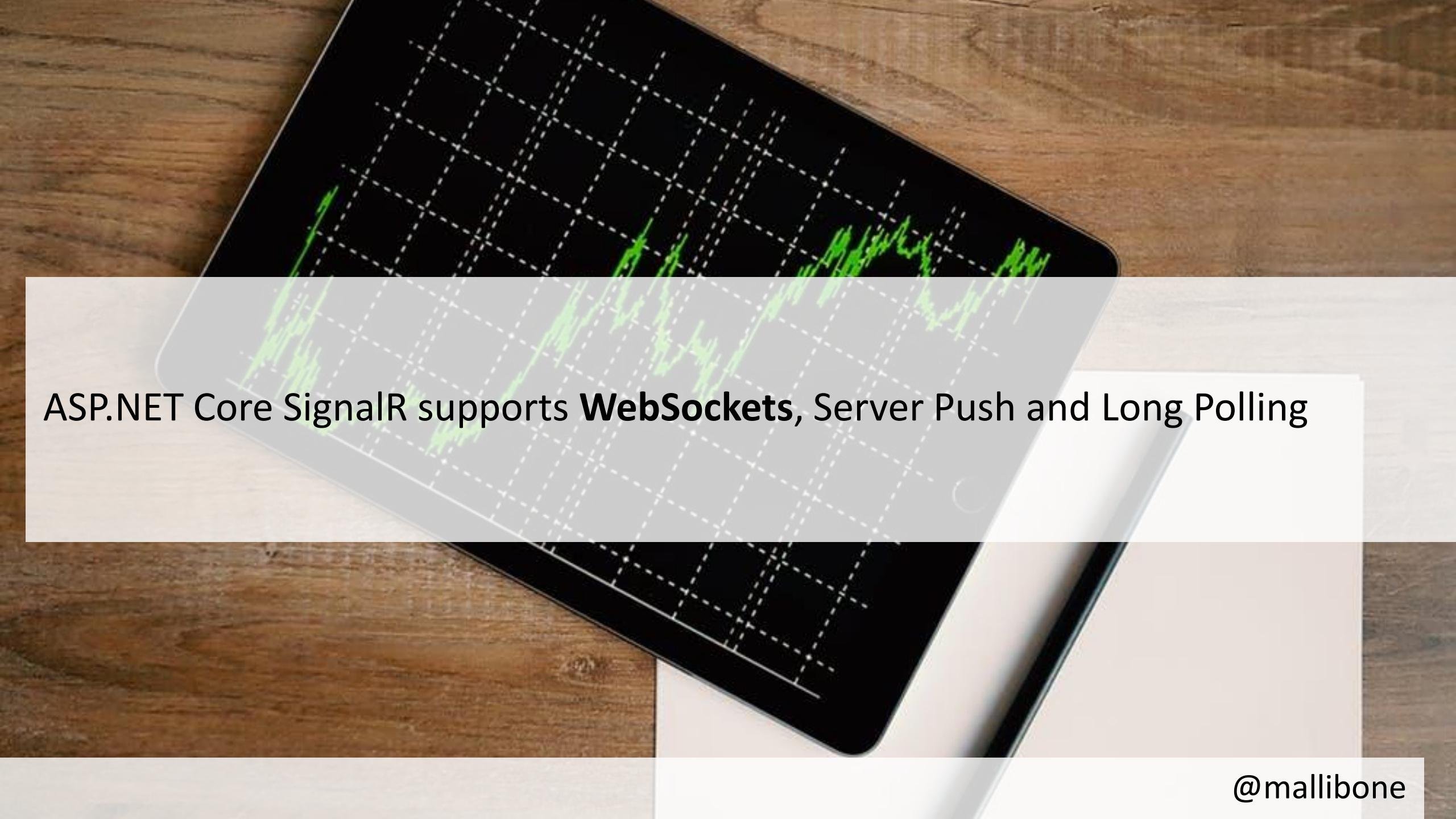


SignalR Service is ASP.Net SignalR Core “carefree”





ASP.NET Core SignalR supports WebSockets, Server Push and Long Polling



ASP.NET Core SignalR supports **WebSockets**, Server Push and Long Polling

```
5
6
7
8
9
10
11     static async Task Main(string[] args)
12     {
13
14         const string backendUrl = "https://iotdemofunction.azurewebsites.net/api/";
15         //const string backendUrl = "http://localhost:7071/api/";
16
17         // Connect to SignalR Service
18         var connect = new HubConnectionBuilder()
19             .WithUrl(backendUrl)
20             .Build();
21
22         // Attach Message handler
23         connect.On<string>("measurement", (messageString) =>
24         {
25             var message = JsonConvert.DeserializeObject<IEnumerable<Measurement>>(messageString);
26             System.Console.WriteLine("Message Received: " + messageString);
27         });
28
29         // Start Monitoring
30         await connect.StartAsync();
31
32         // User input handling
33
34         System.Console.WriteLine("...");
35         System.Console.ReadLine();
36         await connect.DisposeAsync();
37     }
38 }
```

```
5
6
7
8
9
10 static async Task Main(string[] args)
11 {
12     const string backendUrl = "https://iotdemofunction.azurewebsites.net/api/";
13     //const string backendUrl = "http://localhost:7071/api/";
14
15     // Connect to SignalR Service
16     var connect = new HubConnectionBuilder()
17         .WithUrl(backendUrl)
18         .Build();
19
20     // Attach Message handler
21     connect.On<string>("measurement", (messageString) =>
22     {
23         var message = JsonConvert.DeserializeObject<IEnumerable<Measurement>>(messageString);
24         System.Console.WriteLine("Message Received: " + messageString);
25     });
26
27     // Start Monitoring
28     await connect.StartAsync();
29
30     // User input handling
31
32     System.Console.WriteLine("...");
33     System.Console.ReadLine();
34     await connect.DisposeAsync();
35 }
```

```
5
6
7
8
9
10
11     static async Task Main(string[] args)
12     {
13
14         const string backendUrl = "https://iotdemofunction.azurewebsites.net/api/";
15         //const string backendUrl = "http://localhost:7071/api/";
16
17         // Connect to SignalR Service
18         var connect = new HubConnectionBuilder()
19             .WithUrl(backendUrl)
20             .Build();
21
22         // Attach Message handler
23         connect.On<string>("measurement", (messageString) =>
24         {
25             var message = JsonConvert.DeserializeObject<IEnumerable<Measurement>>(messageString);
26             System.Console.WriteLine("Message Received: " + messageString);
27         });
28
29         // Start Monitoring
30         await connect.StartAsync();
31
32         // User input handling
33
34         System.Console.WriteLine("...");
35         System.Console.ReadLine();
36         await connect.DisposeAsync();
37     }
38 }
```

```
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

```
    static async Task Main(string[] args)
{
    const string backendUrl = "https://iotdemofunction.azurewebsites.net/api/";
    //const string backendUrl = "http://localhost:7071/api/";

    // Connect to SignalR Service
    var connect = new HubConnectionBuilder()
        .WithUrl(backendUrl)
        .Build();

    // Attach Message handler
    connect.On<string>("measurement", (messageString) =>
    {
        var message = JsonConvert.DeserializeObject<IEnumerable<Measurement>>(messageString);
        System.Console.WriteLine("Message Received: " + messageString);
    });

    // Start Monitoring
    await connect.StartAsync();

    // User input handling

    System.Console.WriteLine("...");
    System.Console.ReadLine();
    await connect.DisposeAsync();
}
```

.NET STANDARD 2.0

XML

XLinq • XML Document • XPath • XSD • XSL

SERIALIZATION

BinaryFormatter • Data Contract • XML

NETWORKING

Sockets • Http • Mail • WebSockets

IO

Files • Compression • MMF

THREADING

Threads • Thread Pool • Tasks

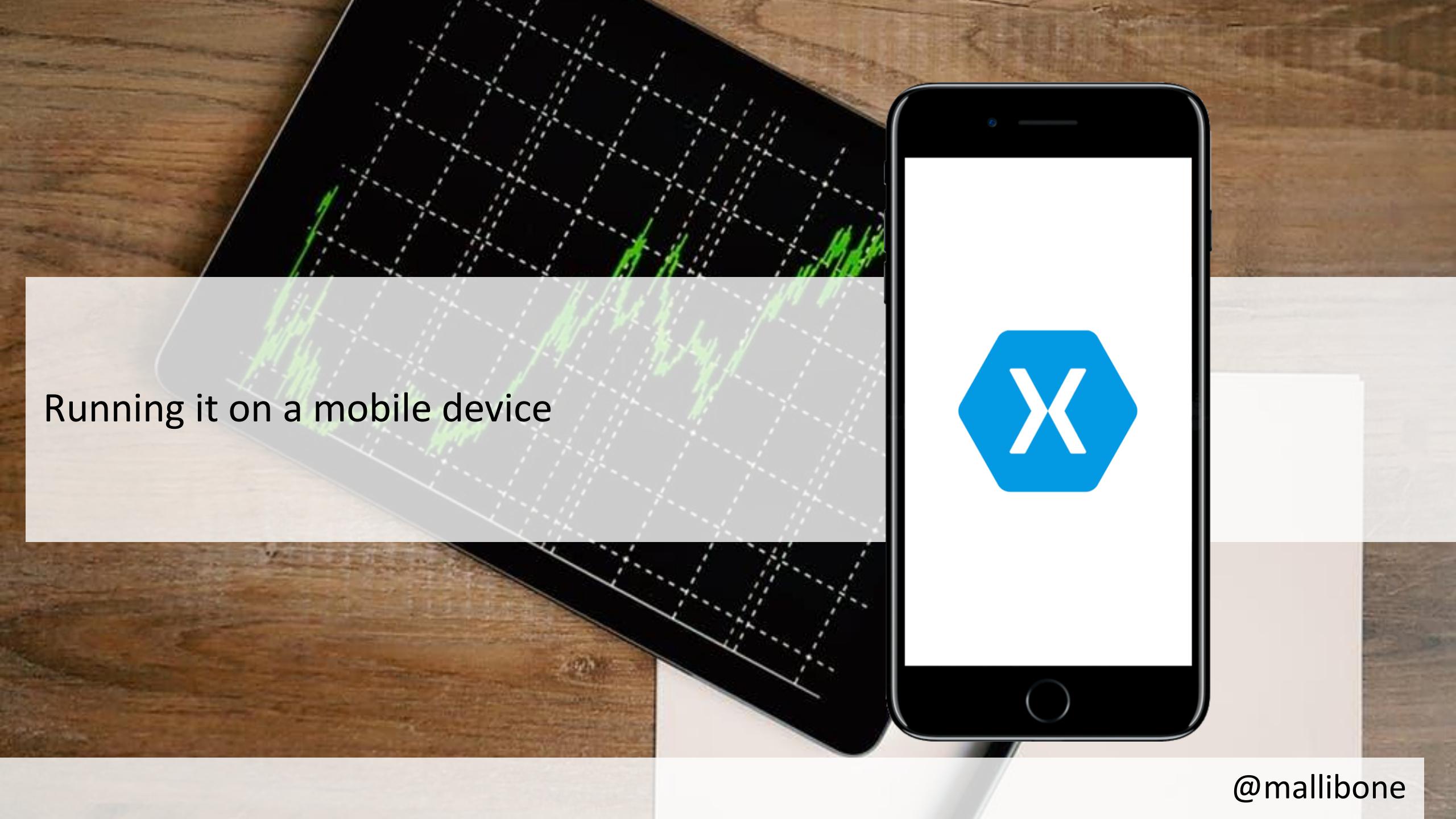
CORE

Primitives • Collections • Reflection • Interop • Linq

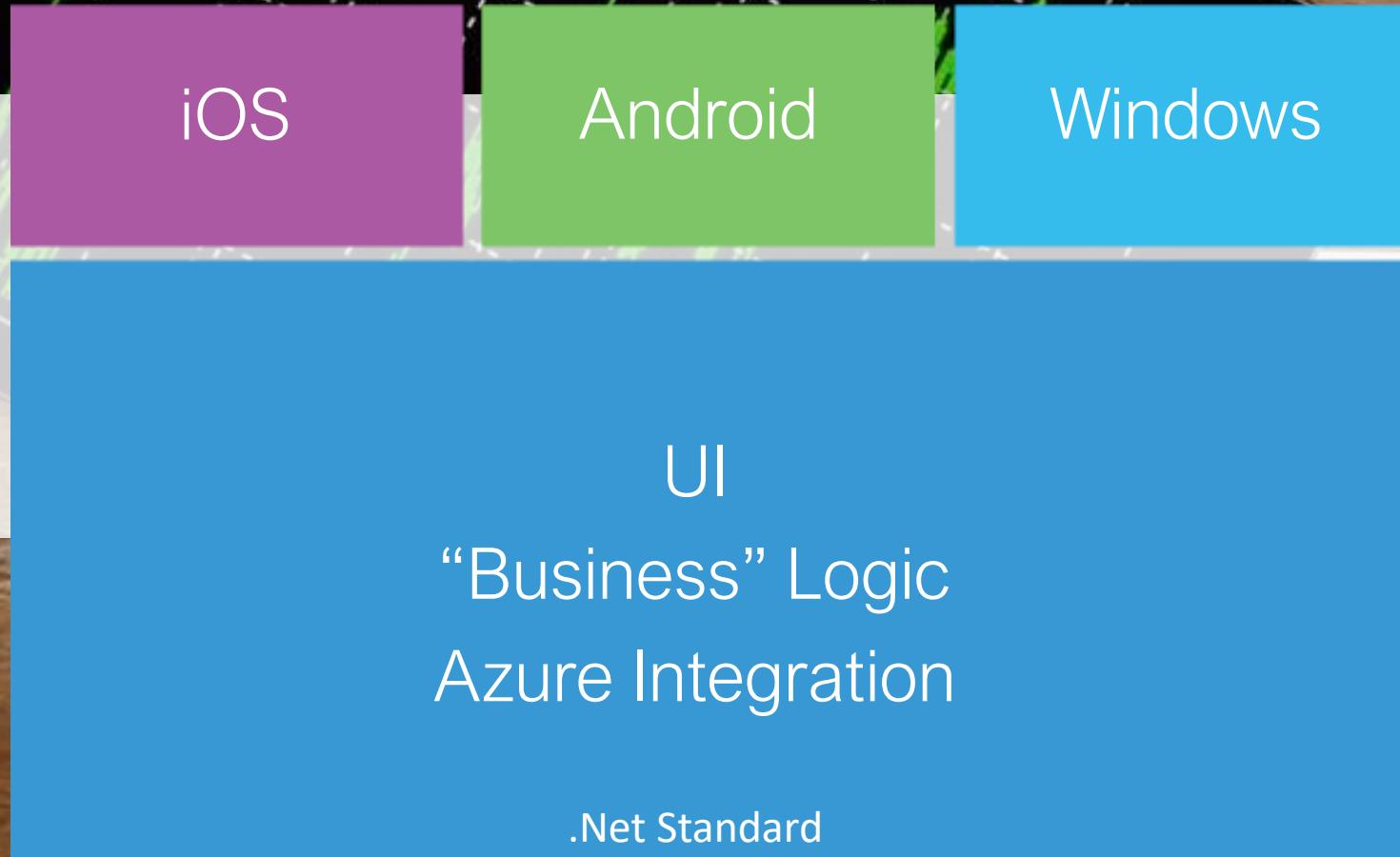


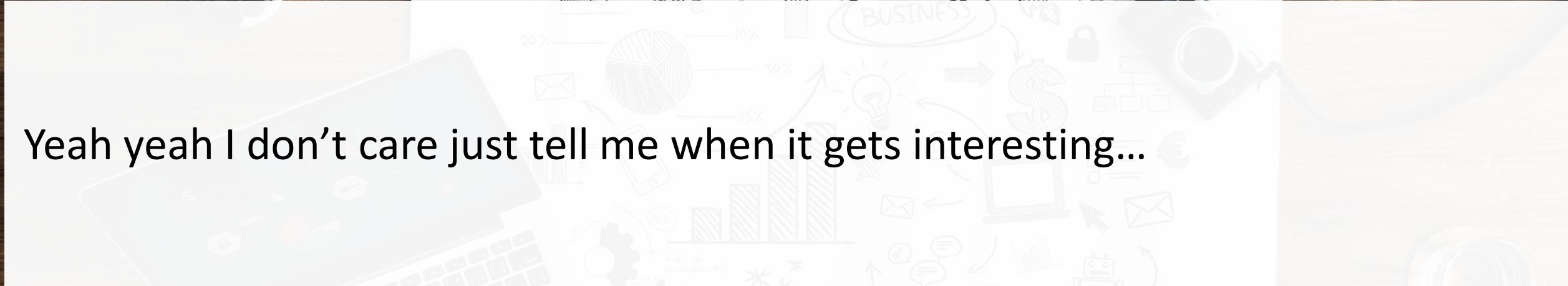
Cross Platform ❤️ .Net Standard

@mallibone



Running it on a mobile device

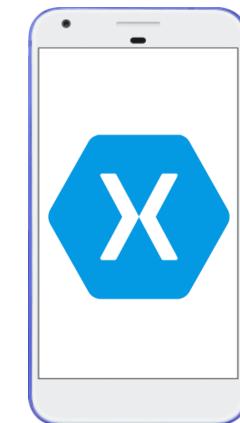
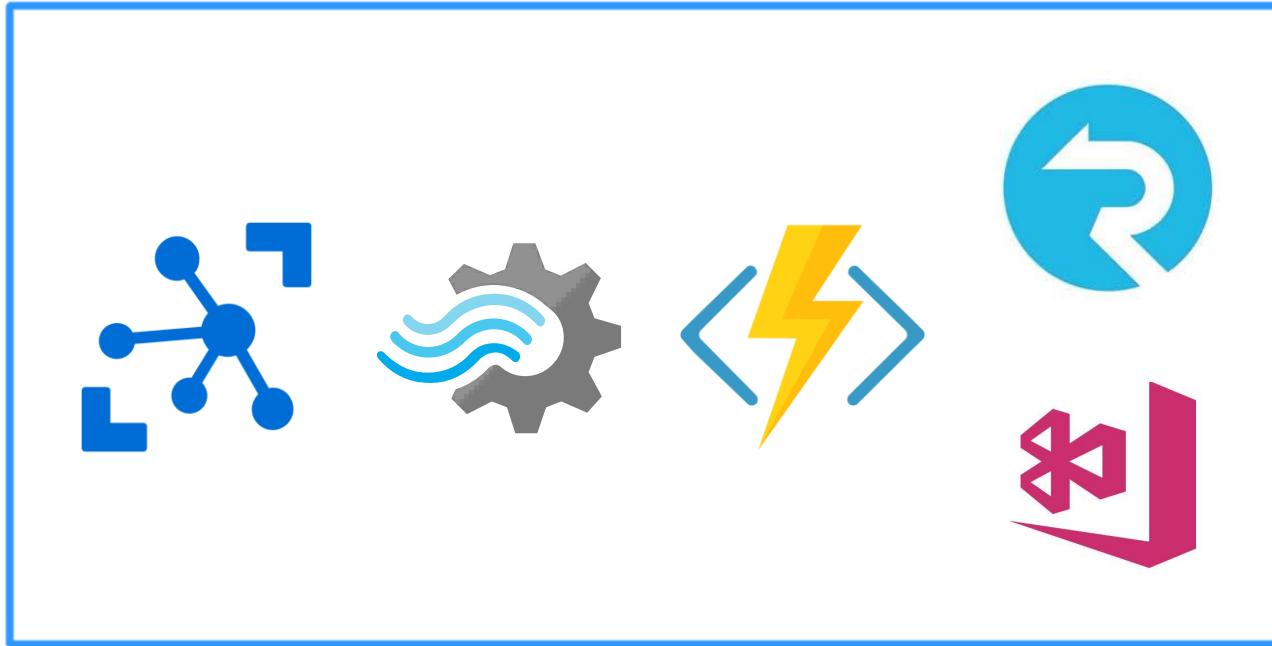
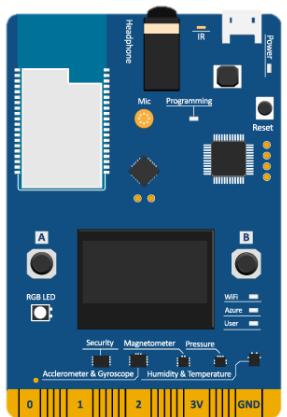




Yeah yeah I don't care just tell me when it gets interesting...



@mallibone



@mallibone



But how do we detect something interesting?



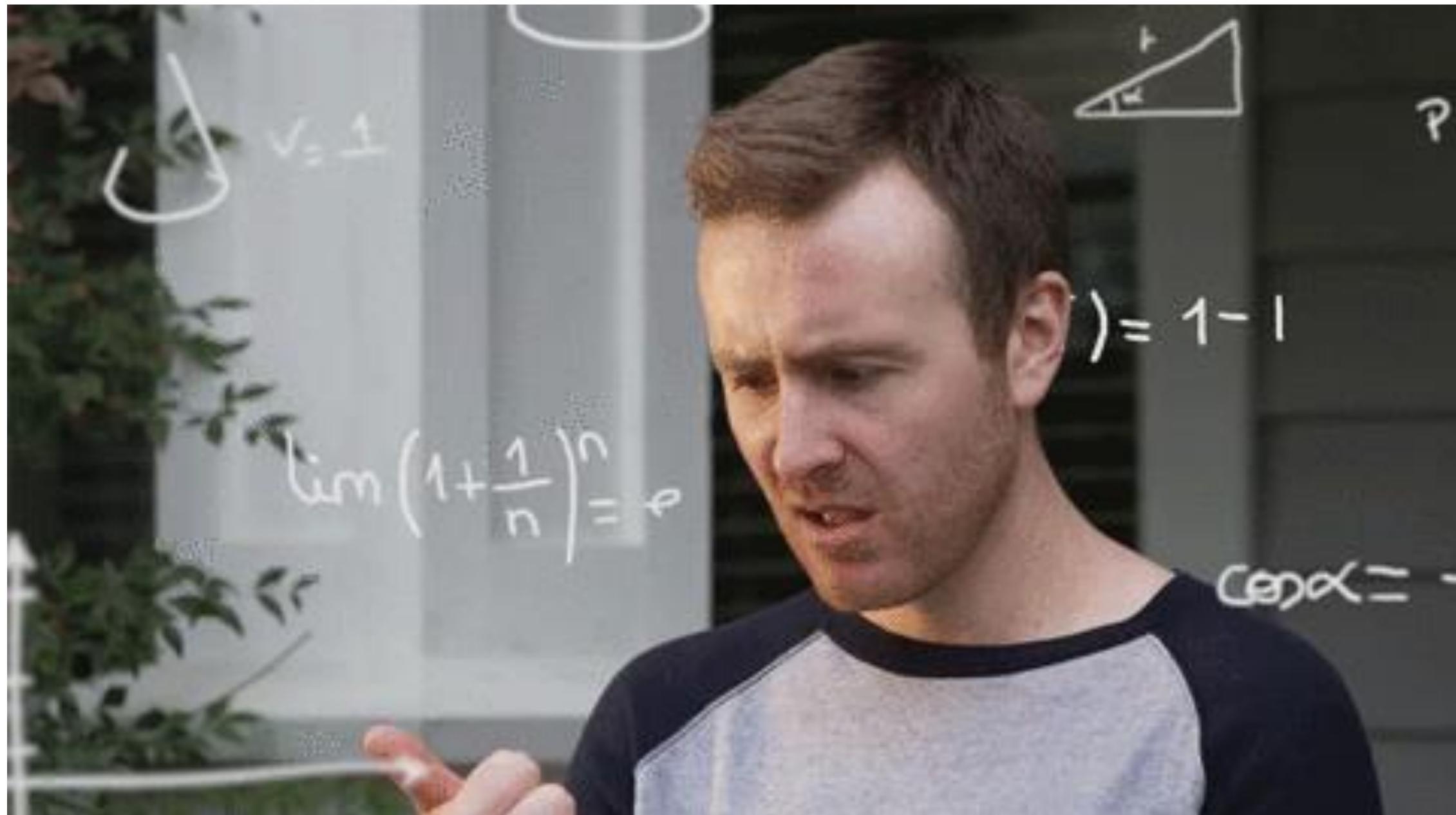
@mallibone



Using machine learning



@mallibone



@mallibone

```

1  WITH AnomalyDetectionStep AS
2  (
3      SELECT
4          EVENTENQUEUEDUTCTIME AS time,
5          CAST(temperature AS float) AS temp,
6          AnomalyDetection_SpikeAndDip(CAST(temperature AS float), 95, 120, 'spikesanddips')
7              OVER(LIMIT DURATION(second, 120)) AS SpikeAndDipScores
8      FROM IoTHub
9  )
10     SELECT
11         time,
12         temp,
13         CAST(GetRecordPropertyValue(SpikeAndDipScores, 'Score') AS float) AS
14         SpikeAndDipScore,
15         CAST(GetRecordPropertyValue(SpikeAndDipScores, 'IsAnomaly') AS bigint) AS
16         IsSpikeAndDipAnomaly
17     INTO AnomalyOutut
18     FROM AnomalyDetectionStep
19
20     SELECT
21         *
22     INTO
23         DataStream
24     FROM
25         IoTHub
26

```

[Input preview](#) [Test results](#)

[Download results](#)

Showing 100 rows from 'anomalyoutut'.

time	temp	spikeanddipscore	isspikeanddipanomaly
"2019-10-25T05:33:44.7720000Z"	20.1	0.39797473087212243	0
"2019-10-25T05:33:43.6770000Z"	20.1	0.39918301431657754	0
"2019-10-25T05:33:42.4270000Z"	20.8	0.46537282782537737	0
"2019-10-25T05:33:41.3790000Z"	20.8	0.46537282782537737	0
"2019-10-25T05:33:40.3310000Z"	20.2	0.41932561007473657	0
"2019-10-25T05:33:39.2690000Z"	20.8	0.46440524496941693	0
"2019-10-25T05:33:38.2060000Z"	20.4	0.4586440721747156	0
"2019-10-25T05:33:37.1580000Z"	31.4	0.015477066118496863	1
"2019-10-25T05:33:35.9390000Z"	20.1	0.4068078323426809	0
"2019-10-25T05:33:34.3920000Z"	20.2	0.4248894379197309	0

@mallbone

```
1 WITH AnomalyDetectionStep AS
2 (
3     SELECT
4         EVENTENQUEUEDUTCTIME AS time,
5         CAST(temperature AS float) AS temp,
6         AnomalyDetection_SpikeAndDip(temperature AS float), 95, 120, 'spikesanddips')
7             OVER(LIMIT DURATION(second, 120)) AS SpikeAndDipScores
8     FROM IoTHub
9 )
10 SELECT
11     time,
12     temp,
13     CAST(GetRecordPropertyValue(SpikeAndDipScores, 'Score') AS float) AS
14     SpikeAndDipScore,
15     CAST(GetRecordPropertyValue(SpikeAndDipScores, 'IsAnomaly') AS bigint) AS
16     IsSpikeAndDipAnomaly
17 INTO AnomalyOutut
18 FROM AnomalyDetectionStep
19
20 SELECT
21     *
22 INTO
23     DataStream
24 FROM
25     IoTHub
26
```

```
1 WITH AnomalyDetectionStep AS
2 (
3     SELECT
4         EVENTENQUEUEDUTCTIME AS time,
5         CAST(temperature AS float) AS temp,
6         AnomalyDetection_SpikeAndDip(temperature AS float), 95, 120, 'spikesanddips')
7             OVER(LIMIT DURATION(second, 120)) AS SpikeAndDipScores
8     FROM IoTHub
9 )
10 SELECT
11     time,
12     temp,
13     CAST(GetRecordPropertyValue(SpikeAndDipScores, 'Score') AS float) AS
14     SpikeAndDipScore,
15     CAST(GetRecordPropertyValue(SpikeAndDipScores, 'IsAnomaly') AS bigint) AS
16     IsSpikeAndDipAnomaly
17 INTO AnomalyOutut
18 FROM AnomalyDetectionStep
19
20 SELECT
21     *
22 INTO
23     DataStream
24 FROM
25     IoTHub
26
```

```
24 FROM  
25 | IoTHub  
26
```

[Input preview](#) [Test results](#)

Showing 100 rows from 'anomalyoutut'.

time	temp	spikeanddipscore	isspikeanddipanomaly
"2019-10-25T05:33:44.7720000Z"	20.1	0.39797473087212243	0
"2019-10-25T05:33:43.6770000Z"	20.1	0.39918301431657754	0
"2019-10-25T05:33:42.4270000Z"	20.8	0.46537282782537737	0
"2019-10-25T05:33:41.3790000Z"	20.8	0.46537282782537737	0
"2019-10-25T05:33:40.3310000Z"	20.2	0.41932561007473657	0
"2019-10-25T05:33:39.2690000Z"	20.8	0.46440524496941693	0
"2019-10-25T05:33:38.2060000Z"	20.4	0.4586440721747156	0
"2019-10-25T05:33:37.1580000Z"	31.4	0.015477066118496863	1
"2019-10-25T05:33:35.9390000Z"	20.1	0.4068078323426809	0
"2019-10-25T05:33:34.3920000Z"	20.2	0.4248894379197309	0

@mallbone

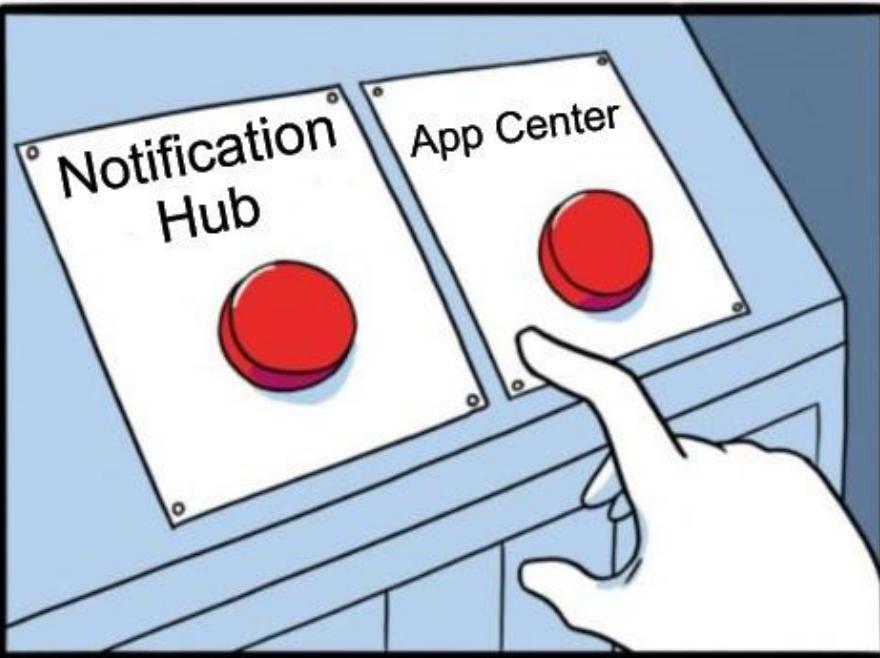


"HOLY FORKING SHIRT!"

#THEGOODPLACE



@mallibone



imgflip.com

JAKE-CLARK.TUMBLR



@mallbone



@mallibone

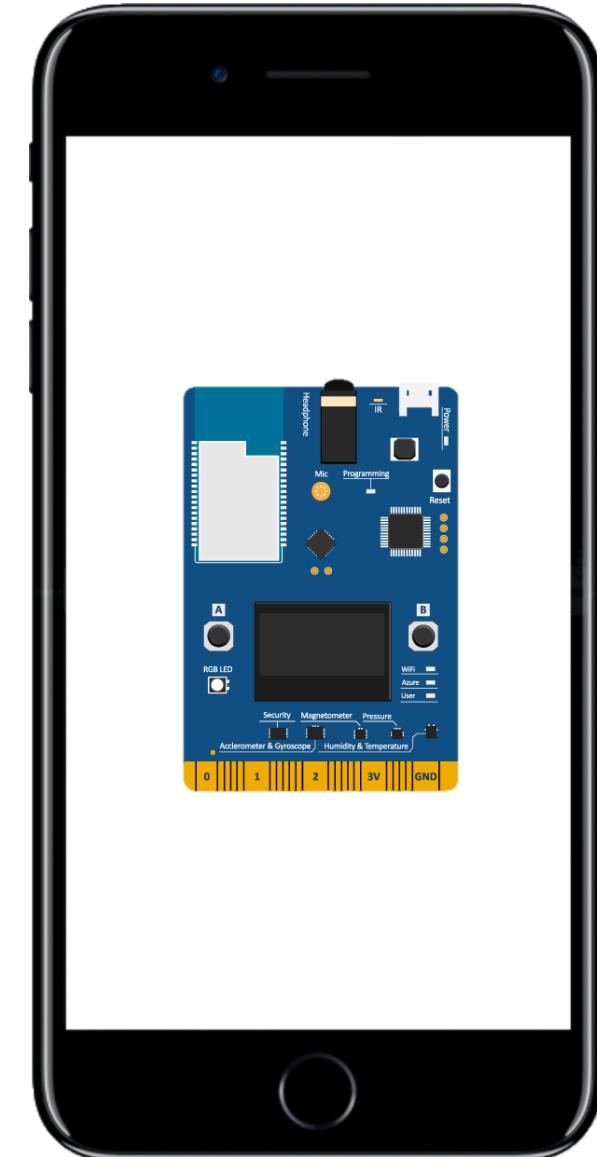


Demo starting at 11:10 😊



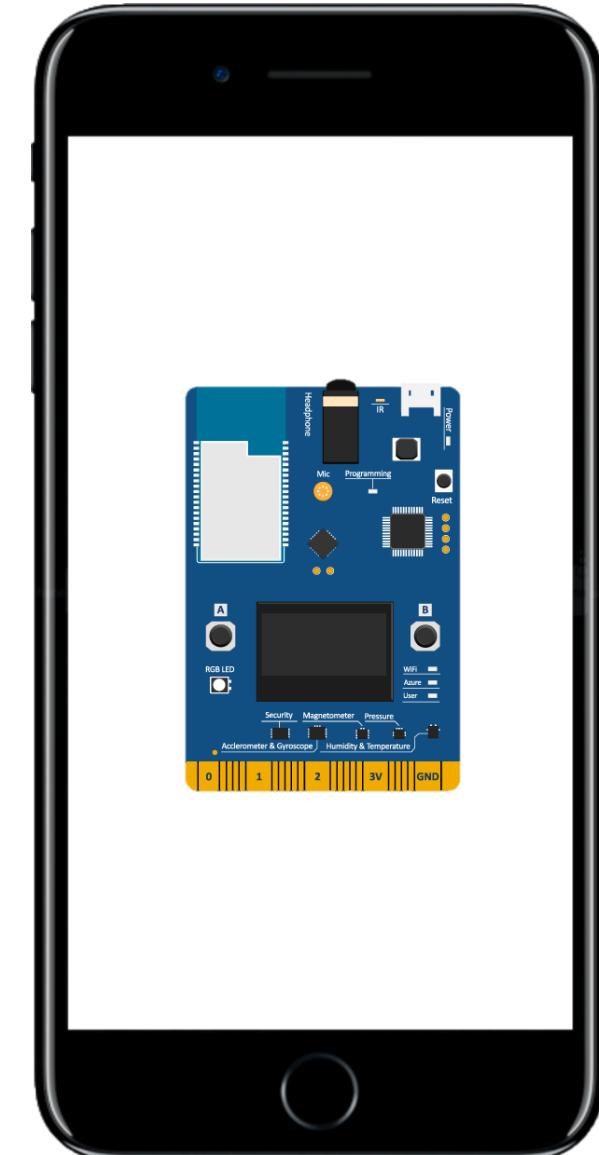
@mallibone

Takeaways



Takeaways

- Mobile is a great IoT monitoring/visualization client
- Live streaming to clients made easy with SignalR
- Implement IoT solutions that can scale with Azure and its services like IoT Hub, Stream Analytics, App Center, ...



Thank you for your time!



 Mark Allibone

 Mobile Lead, Rey Automation AG

 <https://mallibone.com>  FEATURED
COMMUNITY
BLOG

</> <https://bit.ly/2p2GZnm>

 @mallibone

