

# Indoor real-time position tracking using Azure Maps and IoT Central

Deployment Instructions

Clone repo

<https://github.com/Azure-Samples/azuremaps-indoor-realtime-position-tracking>

Create resource group in Azure Subscription

To isolate all resources associated with this deployment, create a new resource group in the Azure portal.

Create IoT Central Application

[Home](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) >

**IoT Central application** ⚙️ ...

Microsoft



**IoT Central application** [♥ Add to Favorites](#)

Microsoft

★ 4.1 (45 Marketplace ratings) | ★ 4.1 (30 external ratings)

Plan

IoT Central application



Create

[Home](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [IoT Central application](#) >

## IoT Central Application ...

**Basics**   Tags   Review + create

Create an IoT Central application with an application template. IoT Central is an IoT app platform that allows you to rapidly build enterprise-grade IoT solutions on a secure, reliable and scalable infrastructure. [Learn more](#)

### Project details

Select the subscription to manage the deployed IoT Central resource and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ	<input type="text" value="IoT_SubscriptionContainer_10"/>
Resource group * ⓘ	<input type="text" value="azuremaps-tracking-rg"/>
	<a href="#">Create new</a>

### Instance details

Resource name *	<input type="text" value="azuremaps-tracking"/>	✓
Application URL *	<input type="text" value="azuremaps-tracking"/>	✓
		azureiotcentral.com
Template * ⓘ	<input type="text" value="Custom application"/>	▼
Region *	<input type="text" value="East US"/>	▼
Pricing plan * ⓘ	<input type="text" value="Standard 2"/>	▼

**Review + create**

< Previous


Next : Tags >

## Add Device (e.g., smart phone)

Go to the IoT Central application URL and add a new device.

+ New

← Import



All devices

Create a new device

×

To create a new device, select a device template, a name, and a unique ID. [Learn more](#)

Device name \*

①

MySmartPhone

Device ID \*

①

MySmartPhone

Organization \*

①

azuremaps-tracking

Device template \*

Unassigned

▼

Simulate this device?

A simulated device generates telemetry that enables you to test the behavior of your application before you connect a real device.

☐ No

Create

Cancel

Select the device that was just created.

Select “Connect”.


Select “QR code”.

Connect

Manage template

Manage device

Devices > MySmartPhone



MySmartPhone

Last data received: N/A | Status: Registered | C

Raw data

Mapped aliases

Files

Timestamp ↓


Message type

Device connection groups

×


ID scope ①

0ne007DDDC3



Device ID ①

MySmartPhone



Choose the connection type for this device. You can change this later if you need to.

Authentication type


Shared access signature (SAS)

▼

Key

QR code

Scan the QR code from a supported mobile app, like [IoT Plug and Play](#), to quickly and securely connect your device to IoT Central.



Close

On your smartphone, install the “IoT Plug and Play” app.

Go to Settings and Registration and select “Scan QR code”.

Then, scan the QR code from the IoT Central application so that your smart phone gets linked to this IoT Central application.

You should be seeing telemetry now in your IoT Central application.

The screenshot shows the Azure IoT Central portal interface. The browser address bar displays the URL: `azuremaps-tracking.azureiotcentral.com/devices/details/MySmartPhone/rawdata`. The page header includes the organization name 'azuremaps-tracking' and a search bar labeled 'Search for devices'. A left-hand navigation menu contains sections: 'Connect' (with sub-items: Devices, Device groups, Device templates), 'Analyze' (with sub-items: Data explorer, Dashboards), 'Manage' (with sub-items: Jobs), and 'Extend'. The main content area is titled 'Devices > MySmartPhone' and shows the device's status as 'Connected' with a green checkmark. It also displays 'Last data received: 10/14/2022, 9:07:09 AM', 'Status: Provisioned', and 'Organization: azuremaps-tracking'. Below this, there are tabs for 'Raw data', 'Mapped aliases', and 'Files'. The 'Raw data' tab is active, showing a table of telemetry data.

Timestamp ↓	Message type	Event creation time	Unmodeled data
> 10/14/2022, 9:06:57 AM	Telemetry		
> 10/14/2022, 9:06:56 AM	Telemetry		
> 10/14/2022, 9:06:56 AM	Telemetry		
> 10/14/2022, 9:06:56 AM	Telemetry		
> 10/14/2022, 9:06:55 AM	Property		

## Create Azure Web PubSub

[Home](#) > [IoT\\_SubscriptionContainer\\_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) >

### Web PubSub Service

Microsoft



#### Web PubSub Service

[Add to Favorites](#)

Microsoft

★ 4.0 (3 Marketplace ratings) | ★ 4.0 (2 external ratings)

Plan

Web PubSub Service

Create

Home > IoT\_SubscriptionContainer\_10 | Resource groups > azuremaps-tracking-rg > Marketplace > Web PubSub Service >

## + Web PubSub Service ...

Web PubSub Service

\* Basics Networking Tags Review + create

Deploy fully managed WebPubSub Service at scale. [Learn more about WebPubSub](#)

### Project Details

Subscription \* ⓘ IoT\_SubscriptionContainer\_10

Resource group \* ⓘ azuremaps-tracking-rg

[Create new](#)

### Service Details

Resource Name \* azuremaps-tracking ✓  
.webpubsub.azure.com

Region \* ⓘ East US

Pricing tier \* ⓘ **Free**  
Up to 20 connections, 40,000 KB messages per day included  
[Change](#)

Review + create

Next : Networking >

[Download a template for automation](#)

Under “Keys”, take a note of the connection string for this service.

## Create Storage Account

Home > IoT\_SubscriptionContainer\_10 | Resource groups > azuremaps-tracking-rg > Marketplace >

## Storage account ✕ ...

Microsoft



### Storage account ♥ Add to Favorites

Microsoft

★ 4.2 (5322 Marketplace ratings) | ★ 4.2 (3548 external ratings)

Plan

Storage account

Create

[Home](#) > [IoT\\_SubscriptionContainer\\_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [Storage account](#) >

## Create a storage account

[Basics](#)   [Advanced](#)   [Networking](#)   [Data protection](#)   [Encryption](#)   [Tags](#)   [Review](#)

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription \*

Resource group \*  [Create new](#)

### Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name ⓘ \*

Region ⓘ \*

Performance ⓘ \*   
☒ **Standard:** Recommended for most scenarios (general-purpose v2 account)   
☐ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ \*

[Review](#)

[< Previous](#)

[Next : Advanced >](#)

Save the value for the connection string under “Access keys” since you will need it later.

Under “Storage browser”, click on “Blob containers”:

Home > [azuremapstracking\\_1665764521994 | Overview](#) > [azuremapstracking](#)

**azuremapstracking | Storage browser** [🔗](#) [☆](#) [...](#)

Storage account

🔍 Search

Overview  
Activity log  
Tags  
Diagnose and solve problems  
Access Control (IAM)  
Data migration  
Events  
Storage browser

azuremapstracking  
Favorites  
Recently viewed  
Blob containers  
\$logs  
View all  
File shares  
Queues  
Tables

+ Add container   ↑ Upload   ↻ Refresh   🗑 Delete   🔒 Change access level   🔄 Restore containers   🛠 Edit columns

Blob containers

🔍 Search containers by prefix   Only show active containers

Showing all 1 items

<input type="checkbox"/>	Name	Last modified	Public access level	Lease state	
<input type="checkbox"/>	\$logs	10/14/2022, 9:22:28 AM	Private	Available	...

Create a new blob container called “iotclogs” (private):

Home > azuremaptracking\_1665764521994 | Overview > azuremaptracking

azuremaptracking | Storage browser

Storage account

Search

Overview  
Activity log  
Tags  
Diagnose and solve problems  
Access Control (IAM)  
Data migration  
Events  
Storage browser  
Data storage  
Containers  
File shares  
Queues  
Tables  
Security + networking  
Networking  
Azure CDN  
Access keys  
Shared access signature

azuremaptracking

Favorites  
Recently viewed  
Blob containers  
\$logs  
View all  
File shares  
Queues  
Tables

+ Add container | Upload | Refresh | Delete | Change access level | Restore container

Blob containers

Search containers by prefix

Showing all 1 items

<input type="checkbox"/>	Name	Last modified	Public
<input type="checkbox"/>	\$logs	10/14/2022, 9:22:28 AM	Private

New container

Name \*  
iotclogs

Public access level ⓘ  
Private (no anonymous access)

Advanced

Create Discard

Create another blob container called “public” with anonymous read access:

Home > azuremaptracking\_1665764521994 | Overview > azuremaptracking

azuremaptracking | Storage browser

Storage account

Search

Overview  
Activity log  
Tags  
Diagnose and solve problems  
Access Control (IAM)  
Data migration  
Events  
Storage browser  
Data storage  
Containers  
File shares  
Queues  
Tables  
Security + networking  
Networking  
Azure CDN  
Access keys  
Shared access signature

azuremaptracking

Favorites  
Recently viewed  
Blob containers  
\$logs  
iotclogs  
View all  
File shares  
Queues  
Tables

+ Add container | Upload | Refresh | Delete | Change access level

Blob containers

Search containers by prefix

Showing all 2 items

<input type="checkbox"/>	Name	Last modified
<input type="checkbox"/>	\$logs	10/14/2022, 9:22:28 AM
<input type="checkbox"/>	iotclogs	10/14/2022, 9:22:28 AM

New container

Name \*  
public

Public access level ⓘ  
Blob (anonymous read access for blobs only)

⚠ Blobs within the container can be read by anonymous request, but container data is not available. Anonymous clients cannot enumerate the blobs within the container.

Advanced

Create Discard

Enable CORS:

Home > azuremaptracking\_1665764521994 | Overview > azuremaptracking

azuremaptracking | Resource sharing (CORS) ☆ ...

Storage account

Search

Save Discard

**Data management**

- Redundancy
- Data protection
- Object replication
- Blob inventory
- Static website
- Lifecycle management
- Azure search

**Settings**

- Configuration
- Data Lake Gen2 upgrade
- Resource sharing (CORS)

CORS is an HTTP feature that enables a web application running under one domain to access resources in another domain. Web browsers implement a security restriction known as same-origin policy that prevents a web page from calling APIs in a different domain. CORS provides a secure way to allow one domain (the origin domain) to call APIs in another domain.

You can set CORS rules individually for each of the storage services (i.e. blob, file, queue, table). Once you set the CORS rules for the service, then a properly authenticated request made against the service from a different domain will be evaluated to determine whether it is allowed according to the rules you have specified.

[Learn more about CORS support for Azure Storage](#)

**Blob service** File service Queue service Table service

Allowed origins	Allowed methods	Allowed headers	Exposed headers	Max age
*	GET			0
	0 selected			0

Install “Azure Storage Explorer”: <https://azure.microsoft.com/en-us/features/storage-explorer/>

Using Azure Storage Explorer, upload the content of the “public” folder in your repo to the corresponding blob container.

Microsoft Azure Storage Explorer

File Edit View Help

EXPLORER

Search for resources

Collapse All Refresh All

- Quick Access
- Emulator & Attached
- Storage Accounts
- Data Lake Storage Gen1 (deprecated)
- IoT\_SubscriptionContainer\_10 (osnaim@microsoft.com)
- Storage Accounts
  - azuremaptracking
    - Blob Containers
      - \$logs
      - iotclogs
      - public
    - File Shares
    - Queues
    - Tables

public

Release Notes: 1.26.0

Upload Download Open Preview New Folder Select All Copy Paste Clone Delete Undo Manage History Folder Statistics Refresh

Active blobs (default) public

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining Days
Woodland_Park_Zoo								
azure-maps-layer-legend.min.css	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	text/css	6.07 KiB	Active	
azure-maps-layer-legend.min.js	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	application/javascript	45.09 KiB	Active	
favicon.ico	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	image/x-icon	14.73 KiB	Active	
Zoo_Map.png	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	image/png	2.10 MiB	Active	

## Create Azure Maps instance

Home > azuremaps-tracking-rg > Marketplace >

### Azure Maps

Microsoft



#### Azure Maps

[Add to Favorites](#)

Microsoft

Plan

Azure Maps

Create



[Home](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [Azure Maps](#) >

## Create an Azure Maps Account resource ...

[Basics](#) [Advanced](#) [Identities](#) [Tags](#) [Review + create](#)

Azure Maps is a collection of geospatial services and SDKs that use fresh mapping data to provide geographic context to web and mobile applications. [Learn more](#)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="IoT_SubscriptionContainer_10"/>
Resource group *	<input type="text" value="azuremaps-tracking-rg"/>
	<a href="#">Create new</a>

### Instance details

Name *	<input type="text" value="azuremaps-tracking"/>
Region *	<input type="text" value="East US"/>
Pricing tier *	<input type="text" value="Gen2 (Maps and Location Insights)"/>

[View full pricing details](#)

### Terms

Azure Maps shares customer-provided address/location queries ("Queries") with third party TomTom for mapping functionality purposes. Queries are not linked to any customer or end-user when shared with TomTom and cannot be

[Review + create](#)

[< Previous](#)

[Next : Advanced >](#)

## Enable CORS:

[Home](#) > [Microsoft.Maps-20221014101719](#) | [Overview](#) > [azuremaps-tracking](#)

azuremaps-tracking | CORS

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Creator

Authentication

Pricing Tier

Identity

CORS

Shared Access Signature

Save Discard

Cross-Origin Resource Sharing (CORS) allows JavaScript code running in a browser on an external host to interact with your backend. Specify the origins that should be allowed to make cross-origin calls (for example: [http://example.com:12345](#)). To allow all, use "\*" or remove all origins from the list. Slashes are not allowed as part of domain or after TLD.

[Learn more](#)

Allowed origins

\*

http://example.com:12345

Save the primary key Azure Maps, which you can find under "Authentication".

# Create Azure Function

[Home](#) > [IoT\\_SubscriptionContainer\\_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) >

## Function App ...

Microsoft



### Function App

 [Add to Favorites](#)

Microsoft

★ 4.1 (4251 Marketplace ratings) | ★ 4.1 (2834 external ratings)

Plan

Function App

Create

[Home](#) > [azuremaps-tracking](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [Function App](#) >

## Create Function App ...

### Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

IoT\_SubscriptionContainer\_10



Resource Group \* ⓘ

azuremaps-tracking-rg

[Create new](#)

### Instance Details

Function App name \*

azuremaps-tracking

.azurewebsites.net

Publish \*



Code



Docker Container

Runtime stack \*

Node.js

Version \*

16 LTS

Region \*

East US

### Operating system

The Operating System has been recommended for you based on your selection of runtime stack.

Operating System \*



Linux



Windows

[Review + create](#)

[< Previous](#)

[Next : Hosting >](#)

And make sure you use the same storage account we created previously:

Home > azuremaps-tracking-rg > Marketplace > Function App >

## Create Function App ...

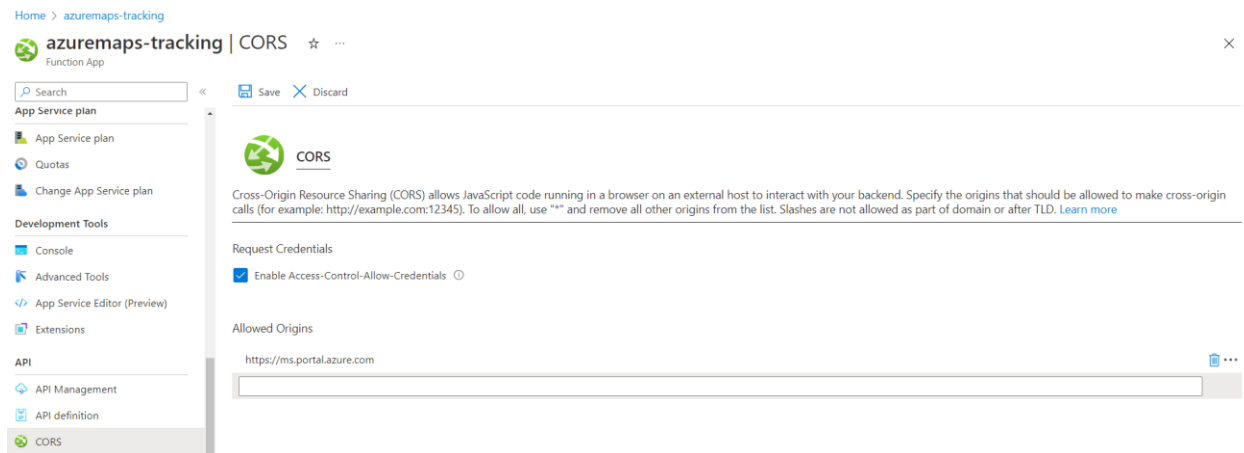
Basics Hosting Networking Monitoring Deployment Tags Review + create

### Storage

When creating a function app, you must create or link to a general-purpose Azure Storage account that supports Blobs, Queue, and Table storage.

Storage account \*  ▼  
[Create new](#)

Once the Function App is created, enable CORS:



Then, under “Configuration”, add a new entry for WebPubSubConnectionString with the corresponding value that we saved previously.

Application settings Function runtime settings General settings

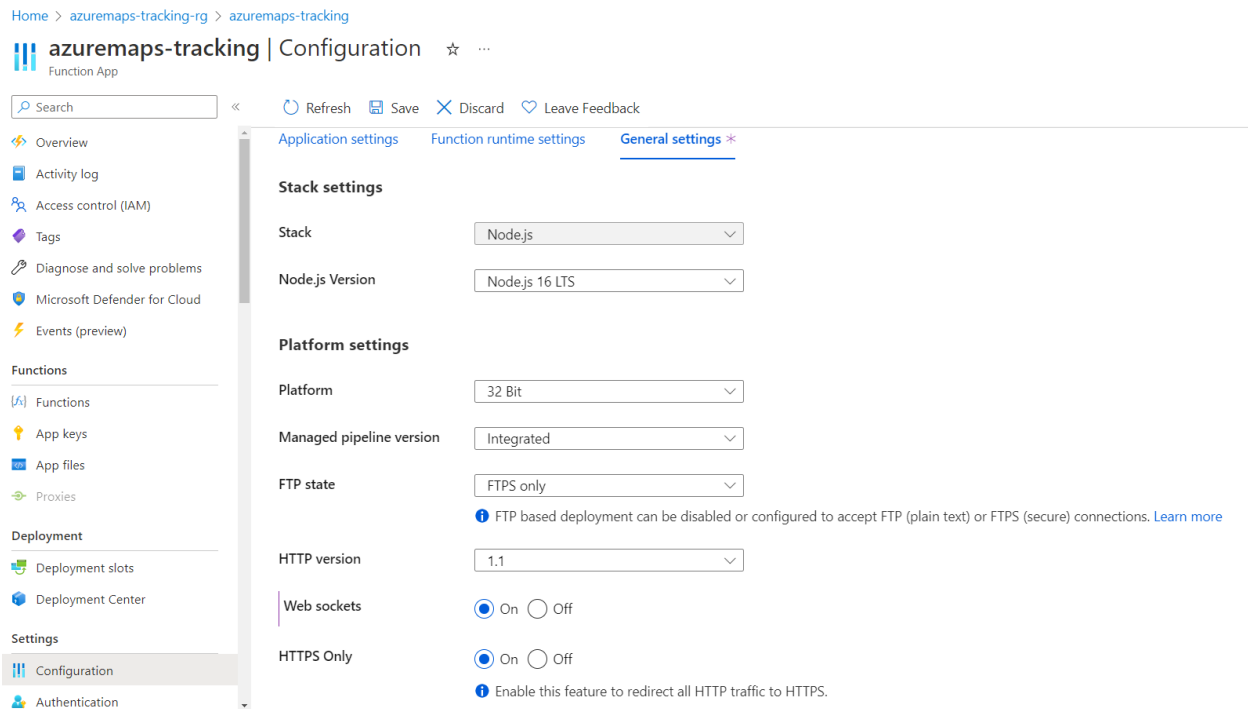
### Application settings

Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below. Application Settings are exposed as environment variables for access by your application at runtime. [Learn more](#)

+ New application setting Show values Advanced edit

Name	Value	Source	Deployment slot setting	Delete	Edit
APPINSIGHTS_INSTRUMENTATIONKEY	<a href="#">Hidden value. Click to show value</a>	App Service Config			
APPLICATIONINSIGHTS_CONNECTION_STRING	<a href="#">Hidden value. Click to show value</a>	App Service Config			
AzureWebJobsStorage	<a href="#">Hidden value. Click to show value</a>	App Service Config			
FUNCTIONS_EXTENSION_VERSION	<a href="#">Hidden value. Click to show value</a>	App Service Config			
FUNCTIONS_WORKER_RUNTIME	<a href="#">Hidden value. Click to show value</a>	App Service Config			
WebPubSubConnectionString	<a href="#">Hidden value. Click to show value</a>	App Service Config			
WEBSITE_CONTENTAZUREFILECONNECTIONSTI	<a href="#">Hidden value. Click to show value</a>	App Service Config			
WEBSITE_CONTENTSHARE	<a href="#">Hidden value. Click to show value</a>	App Service Config			
WEBSITE_NODE_DEFAULT_VERSION	<a href="#">Hidden value. Click to show value</a>	App Service Config			

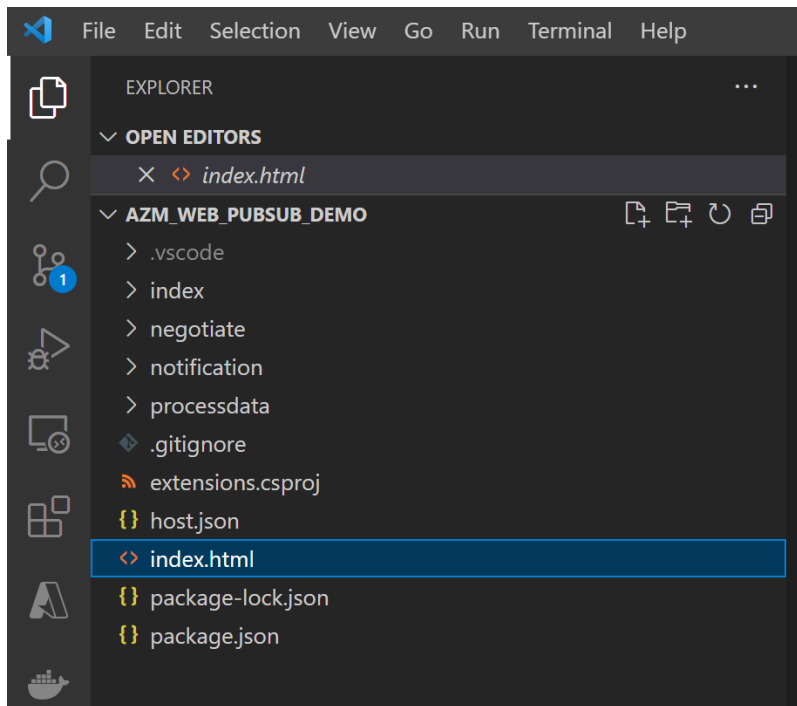
Also, under “Configuration”, enable Web sockets:



## Modify Function App

We will now modify the Azure Function by deploying the code we cloned from the repo.

Open the following folder using VS Code (**AZM\_WEB\_PUBSUB\_DEMO**, which you can find under realtime-azuremaps-update-iotcentraldemo\AzM\_Web\_PubSub\_Demo-v01):



Let's start with "index.html" in the main folder.

First, replace “<YOUR-BLOB-STORAGE-URL>” with the corresponding value, which you can get from Settings/Endpoints/Blob service in the Azure portal. There should be 6 instances of it that you need to replace in this file.

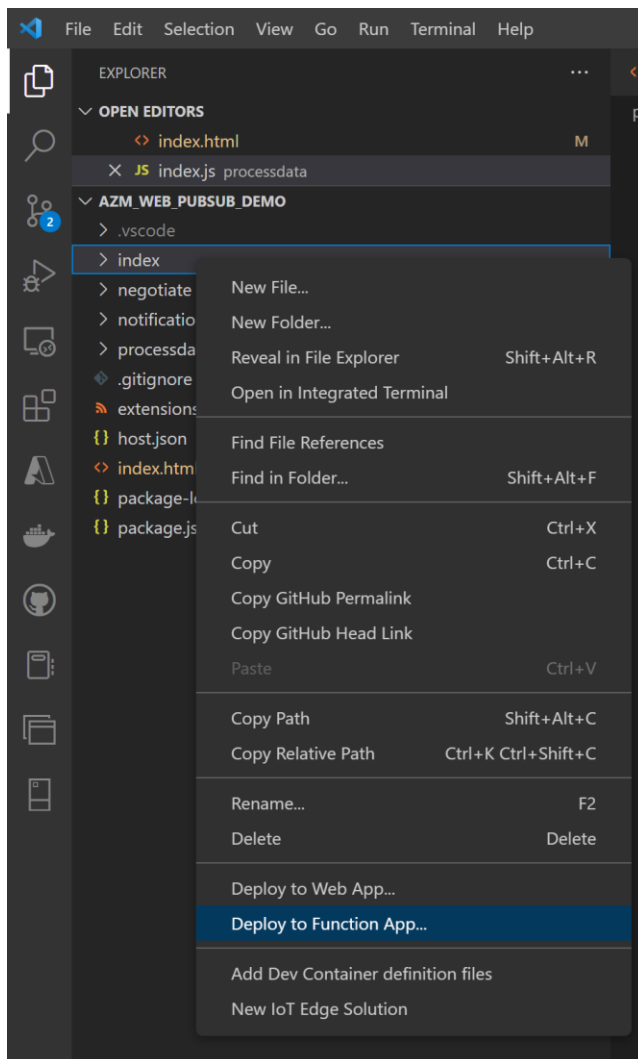
For example:

```
<html>
  <head>
    <title>Azure Maps Web PubSub</title>
    <link rel="shortcut icon" href="https://azuremapstracking.blob.core.windows.net/public/favicon.ico"/>
    <meta charset="utf-8">
```

Then, replace “<YOUR-AZURE-MAPS-KEY>” with the map key you had saved previously.

## Deploy Function App to Azure

We now need to deploy these functions to the Azure Function we created previously. We will do this by right clicking on the corresponding folder and selecting deploy to Function App:



This step will deploy 4 functions to the Function App: index, negotiate, notification, and processdata.

```
TERMINAL  AZURE  JUPYTER  DEBUG CONSOLE  PROBLEMS  OUTPUT
11:11:57 AM azuremaps-tracking: Added app setting WEBSITE_RUN_FROM_PACKAGE to improve performance of function app. Learn more here: https://aka.ms/AA8vxc0
11:12:01 AM azuremaps-tracking: Starting deployment...
11:12:08 AM azuremaps-tracking: Creating zip package...
11:12:17 AM azuremaps-tracking: Zip package size: 2.9 MB
11:12:22 AM azuremaps-tracking: Updating submodules.
11:12:25 AM azuremaps-tracking: Preparing deployment for commit id 'a6e3f01268'.
11:12:29 AM azuremaps-tracking: Skipping build. Project type: Run-From-Zip
11:12:30 AM azuremaps-tracking: Skipping post build. Project type: Run-From-Zip
11:12:30 AM azuremaps-tracking: Triggering recycle (preview mode disabled).
11:12:31 AM azuremaps-tracking: Deployment successful.
11:13:16 AM azuremaps-tracking: Started postDeployTask "npm install (functions)".
11:13:27 AM azuremaps-tracking: Syncing triggers...
11:13:37 AM azuremaps-tracking: Querying triggers...
11:13:40 AM azuremaps-tracking: HTTP Trigger Urls:
  index: https://azuremaps-tracking.azurewebsites.net/api/index
  negotiate: https://azuremaps-tracking.azurewebsites.net/api/negotiate
  notification: https://azuremaps-tracking.azurewebsites.net/api/notification
[Azurite Table Service] [Azurite Queue Service] [Azurite Blob Service] Ln 125, Col 70 Spaces: 4 UTF-8 CRLF HTML
```

Home > azuremaps-tracking-rg > azuremaps-tracking

### azuremaps-tracking | Functions

Function App

Search

+ Create Refresh Delete

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Events (preview)

Functions

Functions

App keys

App files

Your app is currently in read only mode because you are running from a package file. To make any changes update the content in your zip file and WEBSITE\_RUN\_FROM\_PACKAGE app setting.

Filter by name...

<input type="checkbox"/> Name ↑↓	Trigger ↑↓	Status ↑↓
<input type="checkbox"/> index	HTTP	Enabled
<input type="checkbox"/> negotiate	HTTP	Enabled
<input type="checkbox"/> notification	HTTP	Enabled
<input type="checkbox"/> processdata	Timer	Enabled

## Export from IoT Central to Webhook

Create a new Data Export component using the following steps:

Click on “New export” and fill out the fields as follows:

azuremaps-tracking

+ New export

Connect

Data export

← → ↺

azuremaps-tracking.azureiotcentral.com/data-export/exports/create

azuremaps-tracking

Search for devices

☰

Connect

Devices

Device groups

Device templates

Analyze

Data explorer

Dashboards

Manage

Jobs

Extend

Rules

Data export

Security

Audit logs

Permissions

Settings

Application

Customization

IoT Central Home

Save Cancel Rename

Exports > ExportToMap

ExportToMap

Enabled

Data

All of your devices will export data unless you add filters to narrow things down. [Learn more](#)

Type of data to export \*  
Telemetry

Export the data if  
all of the conditions are true

Name \*  
Sensors / Location / Latitude

Operator \*  
Exists

Name \*  
Sensors / Location / Longitude

Operator \*  
Exists

+ Filter + Message property filter

Enrichments

Add additional information to your export. This will appear as a key value pair in exported messages. [Learn more](#)

+ Custom string

+ Property

Add a new Destination:

## New destination

To create a new destination, select a destination type and enter the connection information. [Learn more](#)

Destination name \*

AzureFunction

Destination type \*

Webhook

Callback URL \*

zurewebsites.net/api/notification?clientId=blobs\_extension

Authorization

No auth

### Headers

Add custom headers to each message that you export to this destination. Custom headers can be used by the destination to process the message, apply custom logic, and troubleshoot issues.

+ Header

Create

Cancel

And finally add a Transform using the following transformation query:

```
{
  deviceId: .device.id,
  enqueuedTime: .enqueuedTime,
  telemetry: .telemetry | map({ key: .name, value: .value }) | from_entries,
}
```

## Data transformation

This is an advanced feature, we recommend you start by visiting the [data transformation guide](#). Transformations change the shape of your exported messages into a new format.

1. Add your input message Generate from device template

```
1 {
2   "applicationId": "78d6e626-0242-4a9b-ae23-1e8e724470b7",
3   "enqueuedTime": "2022-10-14T18:45:46.519Z",
4   "device": {
5     "id": "2jdrjb88b90",
6     "name": "Device",
7     "templateId": "urn:iotc:modelDefinition:transform",
8     "templateName": "Device template",
9     "properties": {
10       "reported": {
11         {
12           "id": "urn:iotc:modelDefinition:transform:9k9ajg72v3",
13           "name": "9k9ajg72v3",
14           "value": 81
15         }
16       },
17     },
18     "cloudProperties": [
19       {
20         "id": "urn:iotc:modelDefinition:transform:9sc3ja6s9v",
21         "name": "9sc3ja6s9v",
22         "value": 33
23       }
24     ],
25     "simulated": true,
26     "approved": true,
27     "blocked": false,
28     "provisioned": false,
29     "organizations": [
30       "leaf-organization"
31     ],
32     "organizationPaths": [
33       {
34         "id": "root-organization",
35         "displayName": "Root Organization"
36       }
37     ]
38   }
39 }
```

2. Build transformation query

```
1 # The transformation query specified here will be used to change each ex
2 # message into a different format. You can get started using the exampl
3 # and learn more about the language in documentation:
4 # https://aka.ms/dataexporttransformation
5 {
6   deviceId: .device.id,
7   enqueuedTime: .enqueuedTime,
8   telemetry: .telemetry | map({ key: .name, value: .value }) | from_en
9 }
```

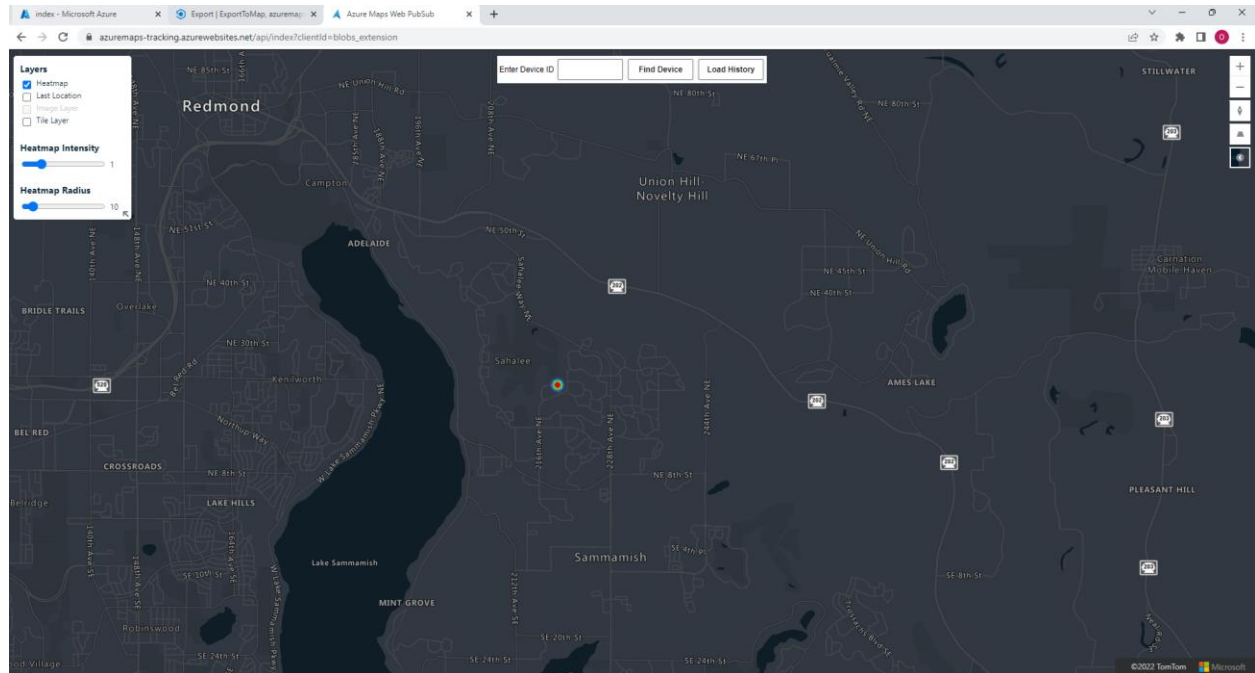
3. Preview output message(s)

```
1 {
2   "deviceId": "2jdrjb88b90",
3   "enqueuedTime": "2022-10-14T18:45:46.519Z",
4   "telemetry": {
5     "279md4vcylz": 47
6   }
7 }
```



Now you can save this Data Export component.

Once this service is showing a “healthy” state, you should be ready to start seeing data points in your map! Just make sure that the IoT Plug and Play application is open in your smartphone.



Just go to the URL that corresponds to the “index” function that you deployed to the Function App:

Home > azuremaps-tracking-rg > azuremaps-tracking | Functions >

**index**  
Function

Search << ☒ Enable ☐ Disable ☐ Delete ☐ Get Function URL ☐ Refresh

**Overview**

**Developer**

- Code + Test
- Integration
- Monitor
- Function Keys

**Get Function Url**

blobs\_extension (system)

Resource group (move) : azuremaps-tracking-rg

Subscription (move) : IoT\_SubscriptionContainer\_10

Subscription ID : e195708a-04a1-404f-bcdc-835381f2576f

## Export from IoT Central to Storage

Let’s create a new Data Export destination for processing the location history.

Using the same Data Export component that we created previously, click on create a new destination with the following details:

## New destination



To create a new destination, select a destination type and enter the connection information. [Learn more](#)

Destination name \*

Destination type \*

Authorization

Connection string \*

Container \* ⓘ

Create

Cancel

We'll use the connection string for the storage account that we saved previously. This will push location history data to iotclogs in blob storage.