Mapping with Alexa

Matthew Garrod
2017 ESRI Developer Summit
Wednesday, March 8
5:30pm - 6:00pm Mesquite B

Mapping with Humans

Alexa
2017 ESRI Developer Summit
Wednesday, March 8
5:30pm - 6:00pm Mesquite B

Agenda

- Overview
- Details
- Demo
- Q&A

Overview

AWS IoT

https://console.aws.amazon.com/iotv2/home

Custom endpoint

This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.

Status

ACTIVE

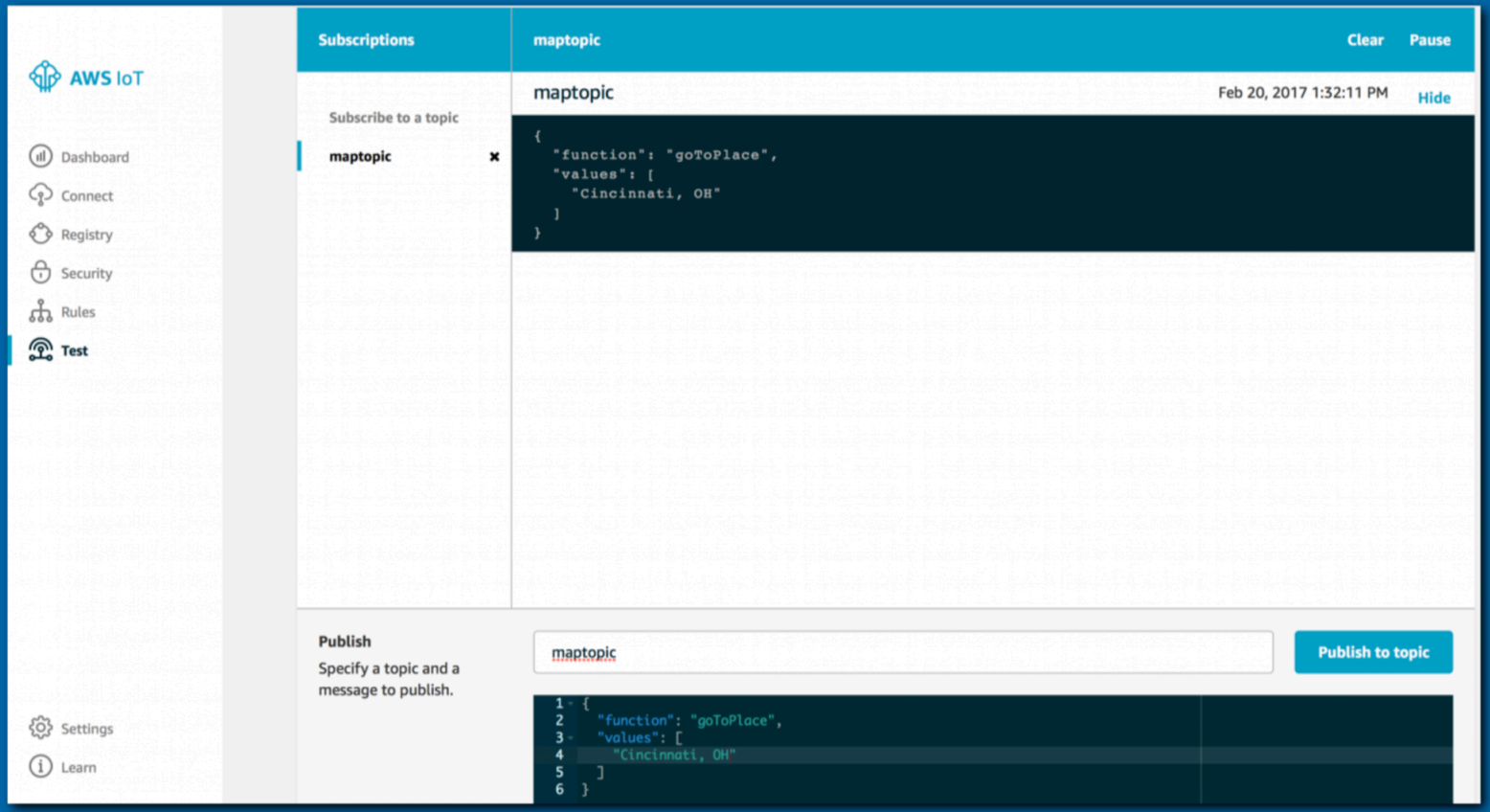
Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.

Endpoint

[redacted].iot.us-east-1.amazonaws.com

AWS IoT

https://console.aws.amazon.com/iotv2/home



AWS IAM

https://console.aws.amazon.com/iam/home

Role #1: lambda_basic_execution_for_iot

Show Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:UpdateThingShadow",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Publish",
            "Resource": "*"
        }
    ]
}
```

Cancel

Role #2: Cognito_mapperUnauth_Role

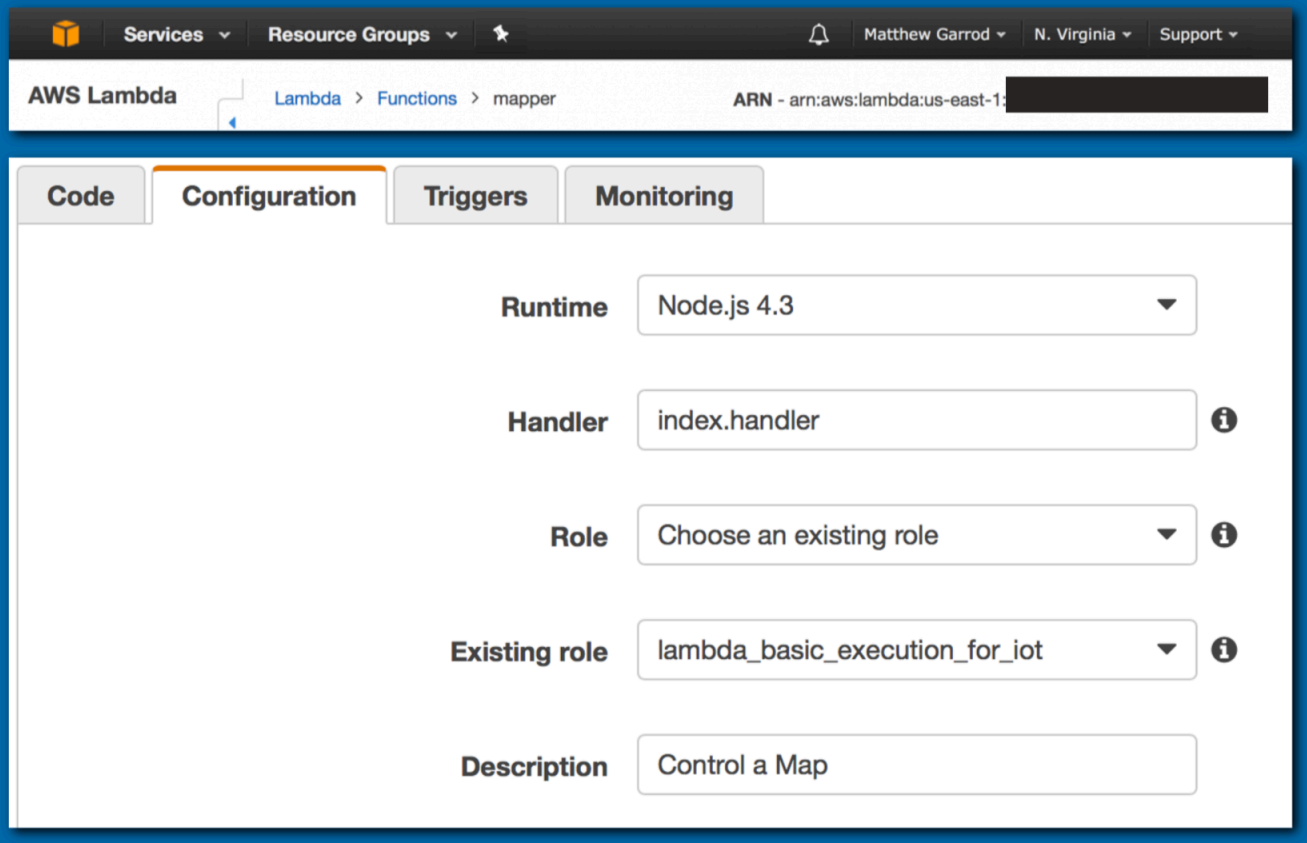
Show Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "mobileanalytics:PutEvents",
                "cognito-sync:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect",
                "iot:Receive",
                "iot:Subscribe"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Cancel

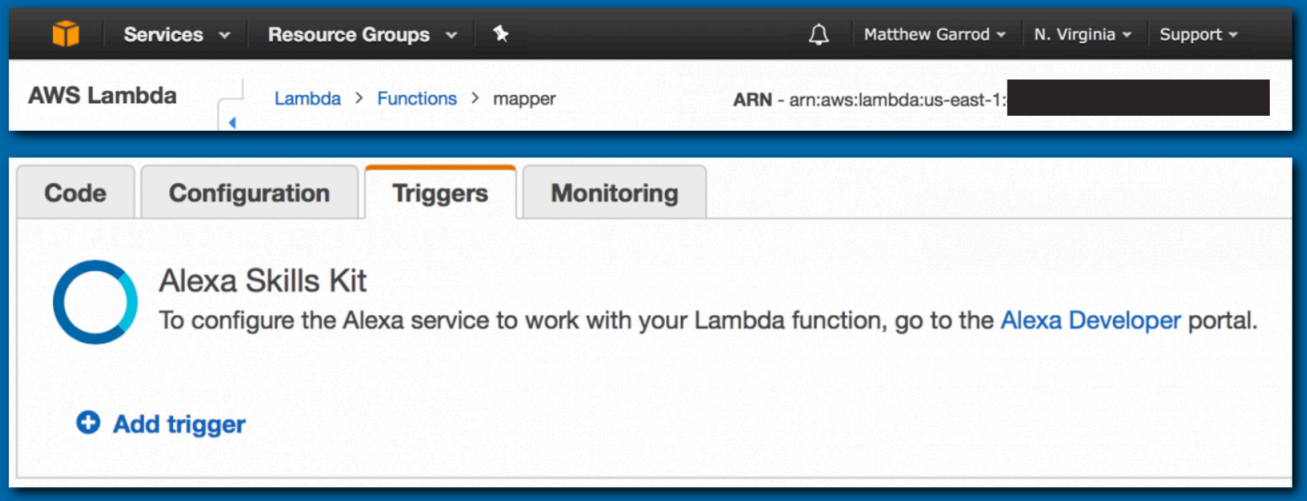
AWS Lambda

https://console.aws.amazon.com/lambda/home



AWS Lambda

https://console.aws.amazon.com/lambda/home



AWS Lambda

https://console.aws.amazon.com/lambda/home

Services | Resource Groups | Matthew Garrod | N. Virginia | Support

AWS Lambda > Lambda > Functions > mapper

ARN - arn:aws:lambda:us-east-1:[redacted]

Code | Configuration | Triggers | Monitoring

Code entry type: Edit code inline

- Edit code inline
- Upload a .ZIP file
- Upload a file from Amazon S3

```
'use strict';
var MapperStatus = require('./

exports.handler = function (event, context) {

    var mapperStatus = new MapperStatus();
    mapperStatus.execute(event, context);

};
```

AWS Lambda

https://console.aws.amazon.com/lambda/home

```
'use strict';
var config = {
    APP_ID: 'amzn1.ask.skill.                                    ',
    IOT_ENDPOINT: '              .iot.us-east-1.amazonaws.com'
};
module.exports = config;
```

```
var iotdata = new AWS.IotData({endpoint: config.IOT_ENDPOINT});
```

AWS Lambda

https://console.aws.amazon.com/lambda/home

```
intentHandlers.ZoomtoPlaceIntent = function (intent, session, response) {

    if (!intent.slots.Place.value && !intent.slots.Country.value) {
        response.ask('Where would you like to go? Say, locate, followed by a place or country');
        return;
    }

    var payload = {
        function: 'goToPlace',
        values: [
            intent.slots.Place.value,
            intent.slots.Country.value
        ]
    };

    var messge = !intent.slots.Place.value ? intent.slots.Country.value : intent.slots.Place.value;

    intentHandlers.publishToIotTopic(session, response, payload, 'locate ' + messge + ' sent to the map');

};
```

```
intentHandlers.publishToIotTopic = function(session, response, payload, tellMessage) {

    if (session.user === undefined || session.user.accessToken === undefined) {
        response.tellWithLinkAccount('Account linking is required for this command. Please use the companion app to enter a topic.');
    }
    else {
        var params = {
            topic: session.user.accessToken.toLowerCase(),
            payload: JSON.stringify(payload),
            qos: 0
        };
        var topicResponse = intentHelper.publishToIotTopic(params);
        topicResponse.then(function(data) {
            response.tell(tellMessage);
        }).catch(function(err) {
            response.tell('Command failed to send to the map.');
        });
    }
};
```

```
publishToIotTopic: function (params) {
    var pub = iotdata.publish(params).promise();
    return pub;
},
```

Alexa Skills Kit

https://developer.amazon.com/edw/home.html#/skills/list

- Skill Information ✓
- Interaction Model ✓
- Configuration ✓
- Publishing Information ✓
- Privacy & Compliance ✓

Skill Type Define a custom interaction model or use one of the predefined skill APIs. Learn more	Custom
Language Language of your skill	English (U.S.)
Application Id The ID for this skill	amzn1.ask.skill.[redacted]
Name Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters.	Web Map
Invocation Name The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...". Invocation Name Guidelines	web map

Alexa Skills Kit

https://developer.amazon.com/edw/home.html#/skills/list

- Skill Information ✓
- Interaction Model ✓
- Configuration ✓
- Publishing Information ✓
- Privacy & Compliance ✓

Intent Schema

```
{
  "intents": [
    {
      "intent": "MoveMapIntent",
      "slots": [
        {
          "name": "PanDirection",
          "type": "PAN_DIRECTIONS"
        },
        {
          "name": "ZoomDirection",
```

Sample Utterances

```
MoveMapIntent to pan {PanDirection}
MoveMapIntent to zoom {ZoomDirection}

ZoomtoPlaceIntent to locate a place
ZoomtoPlaceIntent to locate {Place}
ZoomtoPlaceIntent to locate a country
ZoomtoPlaceIntent to locate {Country}

TiltCameraIntent to tilt the camera
TiltCameraIntent to tilt the camera {Degree} degrees
TiltCameraIntent angle of {Degree}
```

Slots

Type	Values
PAN_DIRECTIONS	up | down | left | right
ZOOM_DIRECTIONS	in | out

Alexa Skills Kit

https://developer.amazon.com/edw/home.html#/skills/list

- Skill Information ✓
- Interaction Model ✓
- Configuration ✓
- Publishing Information ✓
- Privacy & Compliance ✓

Endpoint

Service Endpoint Type:

- (•) **AWS Lambda ARN (Amazon Resource Name)** ℹ — *Recommended*
  AWS Lambda is a server-less compute service that runs your code in response to events and automatically manages the underlying compute resources for you.
  More info about AWS Lambda
  How to integrate AWS Lambda with Alexa
- ( ) **HTTPS**

Pick a geographical region that is closest to your target customers: ℹ

- [x] North America
- [ ] Europe

North America

arn:aws:lambda:us-east-1[redacted]

Account Linking

Do you allow users to create an account or link to an existing account with you?
Learn more

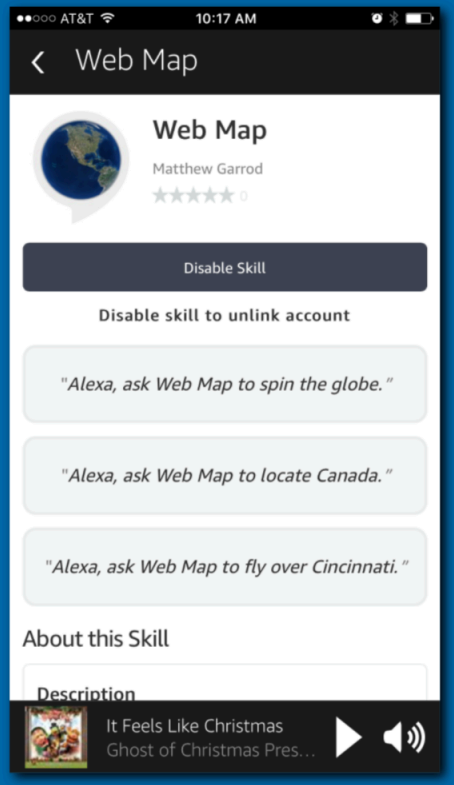
(•) Yes ( ) No

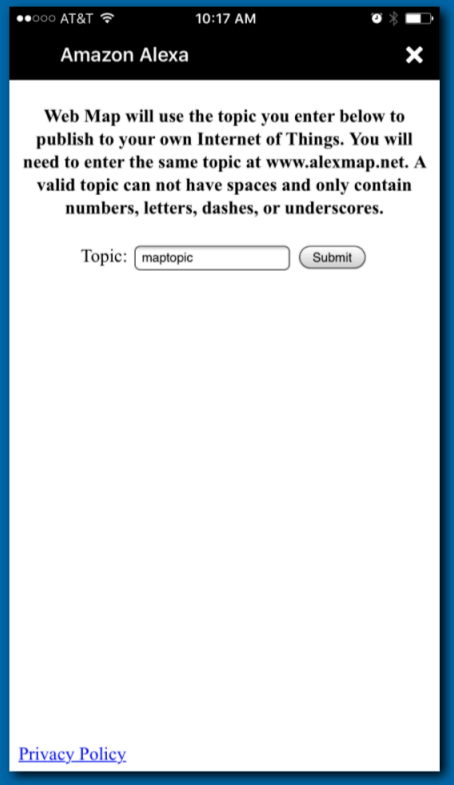
Authorization URL
The url where customers will be redirected in the companion app to enter login credentials.

https://www.alexamap.net/login.html

Alexa Skills Kit

https://developer.amazon.com/edw/home.html#/skills/list





AWS IAM

https://console.aws.amazon.com/iam/home

Role #1: lambda_basic_execution_for_iot

Show Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:UpdateThingShadow",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Publish",
            "Resource": "*"
        }
    ]
}
```

Cancel

Role #2: Cognito_mapperUnauth_Role

Show Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "mobileanalytics:PutEvents",
                "cognito-sync:*"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect",
                "iot:Receive",
                "iot:Subscribe"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Cancel

AWS Congnito

https://console.aws.amazon.com/cognito/home

Edit identity pool

From this page you can modify the details of your identity pool. An identity pool must have a unique name and a set of authenticated and unauthenticated roles. The roles are saved with your identity pool and whenever we receive a request to authorize a user we will automatically utilize the roles you specify here. You will be required to specify the identity pool id from this page when initializing the Amazon Cognito client SDK. Learn more about using IAM roles with Amazon Cognito.

Identity pool name* web map

Identity pool ID us-east-1[redacted]

Unauthenticated role Cognito_mapperUnauth_Role Create new role

Authenticated role Cognito_mapperAuth_Role Create new role

Unauthenticated identities

Amazon Cognito can support unauthenticated identities by providing a unique identifier and AWS credentials for users who do not authenticate with an identity provider. If your application allows customers to use the application without logging in, you can enable access for unauthenticated identities. Learn more about unauthenticated identities.

☑ Enable access to unauthenticated identities

AWS IoT Device JS SDK w/ MQTT.js

https://github.com/aws/aws-iot-device-sdk-js

- Install Node.js and npm
- $ git clone https://github.com/aws/aws-iot-device-sdk-js.git
- $ cd aws-iot-device-sdk-js
- $ export AWS_SERVICES=cognitoidentity
- $ sudo npm install -g browserify
- $ npm run-script browserize
- $ cp browser/aws-iot-sdk-browser-bundle.js

```
<script src="js/aws-iot-sdk-browser-bundle.js"></script>
```

aws-iot-sdk-browser-bundle package.json

```
{
  "name": "aws-iot-sdk-browser-bundle",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"bundle exists\" && exit 0"
  },
  "author": "",
  "license": "Apache-2.0",
  "dependencies": {
    "aws-iot-device-sdk": "^1.0.11",
    "aws-sdk": "^2.3.0"
  }
}
```

aws-iot-device-sdk package.json

```
  "dependencies": {
    "mqtt": "1.11.2",
    "minimist": "1.2.0",
    "websocket-stream": "^3.1.0",
    "crypto-js": "3.1.6"
  },
```

AWS IoT Device JS SDK w/ MQTT.js

https://github.com/aws/aws-iot-device-sdk-js

```
var AWS = awsrequire('aws-sdk');
var AWSIoTData = awsrequire('aws-iot-device-sdk');
```

```
//
// Create a client id to use when connecting to AWS IoT.
//
var clientId = 'mqtt-explorer-' + (Math.floor((Math.random() * 100000) + 1));

AWS.config.region = 'us-east-1';
AWS.config.credentials = new AWS.CognitoIdentityCredentials({
    IdentityPoolId: config.AWS_POOLID
});

const mqttClient = AWSIoTData.device({
    //
    // Set the AWS region we will operate in.
    //
    region: AWS.config.region,
    //
    // Use the clientId created earlier.
    //
    clientId: clientId,
    //
    // Connect via secure WebSocket
    //
    protocol: 'wss',
    //
    // Set the maximum reconnect time to 8 seconds; this is a browser application
    // so we don't want to leave the user waiting too long for reconnection after
    // re-connecting to the network/re-opening their laptop/etc...
    //
    maximumReconnectTimeMs: 8000,
    //
    // Enable console debugging information (optional)
    //
    debug: false,
    //
    // IMPORTANT: the AWS access key ID, secret key, and sesion token must be
    // initialized with empty strings.
    //
    accessKeyId: '',
    secretKey: '',
    sessionToken: ''
});
```

AWS IoT Device JS SDK w/ MQTT.js

https://github.com/aws/aws-iot-device-sdk-js

```
//
// Attempt to authenticate to the Cognito Identity Pool.  Note that this
// example only supports use of a pool which allows unauthenticated
// identities.
//
window.mqttConnectionHandler = function() {

    var cognitoIdentity = new AWS.CognitoIdentity();
    AWS.config.credentials.get(function(err, data) {
        if (!err) {
            console.log('retrieved identity: ' + AWS.config.credentials.identityId);
            var params = {
                IdentityId: AWS.config.credentials.identityId
            };
            cognitoIdentity.getCredentialsForIdentity(params, function(err, data) {
                if (!err) {
                    //
                    // Update our latest AWS credentials; the MQTT client will use these
                    // during its next reconnect attempt.
                    //
                    mqttClient.updateWebSocketCredentials(data.Credentials.AccessKeyId, data.Credentials.SecretKey, data.Credentials.SessionToken, 21600000);

                } else {
                    console.log('error retrieving credentials: ' + err);
                    alert('error retrieving credentials: ' + err);
                }
            });
        } else {
            console.log('error retrieving identity:' + err);
            alert('error retrieving identity: ' + err);
        }
    });
}
window.mqttConnectionHandler();
```

AWS IoT Device JS SDK w/ MQTT.js

https://github.com/aws/aws-iot-device-sdk-js

```
//
// Connect handler;
// Subscribe to lifecycle events on the first connect event.
//
window.mqttClientConnectHandler = function() {
    console.log('connect');
    reconnectCounter = 0;
    //
    // Subscribe to our current topic.
    //
    mqttClient.subscribe(currentlySubscribedTopic);
};
```

```
//
// Message handler for lifecycle events;
// connect/disconnect.
//
window.mqttClientMessageHandler = function(topic, payload) {

    var jsonPayload = JSON.parse(payload);

    // call to the map
    mapFunctions.callMapFunction(jsonPayload);
};
```

```
//
// Install connect/reconnect event handlers.
//
mqttClient.on('connect', window.mqttClientConnectHandler);
mqttClient.on('reconnect', window.mqttClientReconnectHandler);
mqttClient.on('message', window.mqttClientMessageHandler);
```

ArcGIS API for JS 4.2

https://developers.arcgis.com/javascript

maptopic

```
{
  "function": "goToPlace",
  "values": [
    "Cincinnati, OH"
  ]
}
```

```
callMapFunction: function(jsonPayload) {

    // call the passed function
    var func = lang.hitch(this, jsonPayload.function, jsonPayload.values);
    func();
},
```

```
// Alexa function to go to an address or country
goToPlace: function(params) {
    var placeToGo = params[0] !== undefined && params[0] != null ? params[0] : params[1];
    this.updateStatus("going to " + placeToGo, true);

    var address = {
       "SingleLine": placeToGo
    };
    var addressParams = {address: address};
    this.locator.addressToLocations(addressParams).then(lang.hitch(this, function(addresses) {

        if (addresses.length > 0) {

            this.view.goTo({
                target: new Extent(addresses[0].extent.xmin, addresses[0].extent.ymin, addresses[0
                heading: this.view.camera.heading
            }).then(lang.hitch(this, function() {
                this.updateStatus("command complete", false, 3000);
            }));
```

Demo

Q&A

https://github.com/mgarrod