

IBM Watson IoT – T1 – MQTT Client

Käyttöönottoehtävä 1

Tässä käyttöönottoehtävässä tarvitaan vain tavallinen Windows-tietokone, johon sinulla on asennusoikeudet, sekä verkkoyhteys Internetiin.

1. Harjoitusympäristö – IBM Bluemix Watson IoT

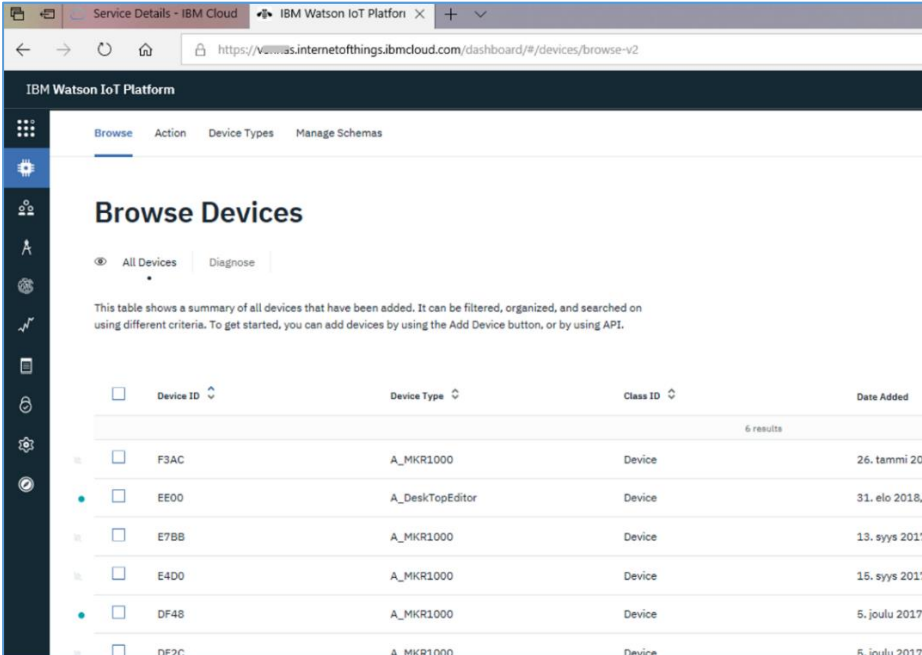
Erillisen ohjeen mukaisesti luot itsellesi IBM –käyttäjätunnuksen ja perustat itsellesi ”organisaation” ja luot kaksi IoT – laitetta.

Ohje on nimeltään ”IoT IBM Watson MKR1000 Aloitus Getting Started käyttöönottoehtävä 0”

Ohje kertoo, kuinka luot laitteeksi Arduino MKR1000 –kehityskortin. Tällä kertaa luomme kuitenkin ”laitteen”, jonka roolia voimme pelata lähes miltä hyvänsä Windows-, Android- tai iOS –ympäristöön tarjolla olevasta MQTT client -sovelluksesta.

Luo ”laite” tai kaksi soveltaen edellä mainittua ohjetta. Anna laitteelle tyyppi, josta tunnistat laitteen olevan vain tietokoneelta mqtt client -sovelluksella hallittava laite. Tyyppi voi olla esim: ”A_DeskTop_Editor”. Muista tallentaa varmaan paikkaan luomasi Organization ID, Device Type, Device ID, Authentication method, Authentication token! ON ERYTÄISEN TÄRKEÄÄ TALLENTAAMIN TEKSTITIEDOSTOON AUTHENTICATION TOKEN!

Kuva 1.1 Yksi Desk Top -sovellus ja useita todellisia laitteita nähtävissä IBM Watson -laiteluettelossa. Vihreällä pisteellä merkityillä on tällä hetkellä yhteys.



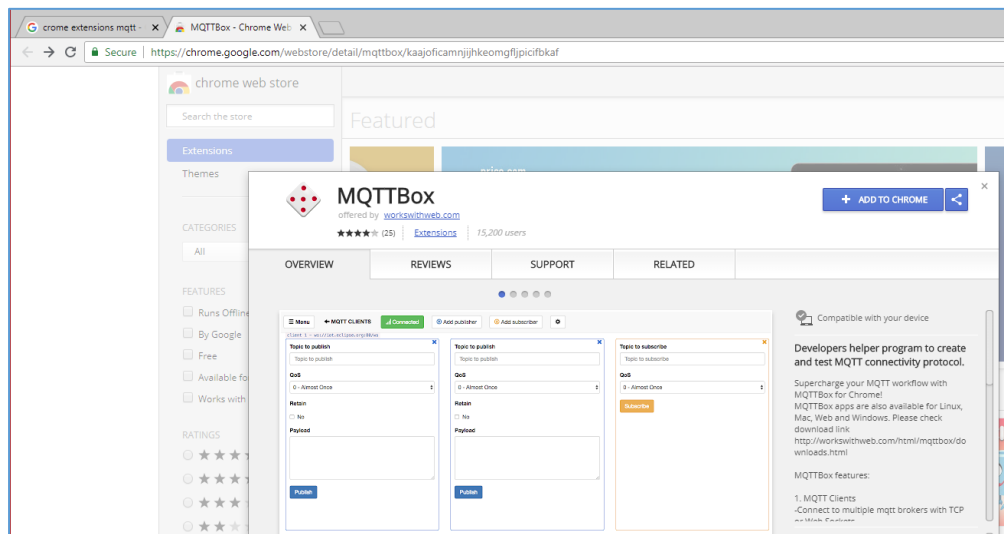
Device ID	Device Type	Class ID	Date Added
F3AC	A_MKR1000	Device	26. tammi 2018
EE00	A_DeskTopEditor	Device	31. elo 2018, 17
E7BB	A_MKR1000	Device	13. syys 2017,
E4D0	A_MKR1000	Device	15. syys 2017,
DF48	A_MKR1000	Device	5. joul 2017,
DF2C	A_MKR1000	Device	5. joul 2017,

2. MQTT broker:in toiminnan tarkasteluun voidaan käyttää selainlaajennusta, Windows-sovellusta, Android-sovellusta tai iOS-sovellusta

2.1 Sovelluksen valinta

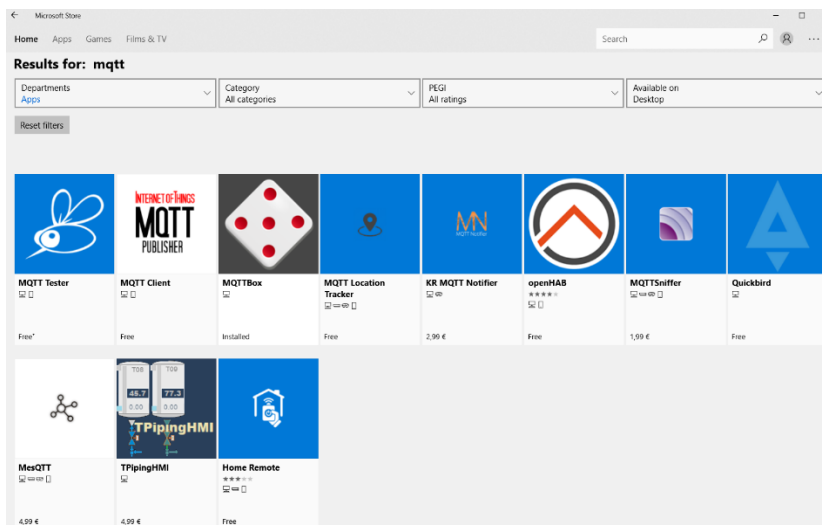
Internet-selainohjelmiin voidaan asentaa laajennuksia eri tarkoituksia varten. Avaa selain ja etsi ”browser extension mqtt”. Tuloksena pitäisi tulla linkki esimerkiksi kohteeseen MQTTBox Chrome Web Store. Voit avata linkin ja asentaa laajennuksen.

Kuva 2.1.1 MQTTBox selainlaajennus / 2018

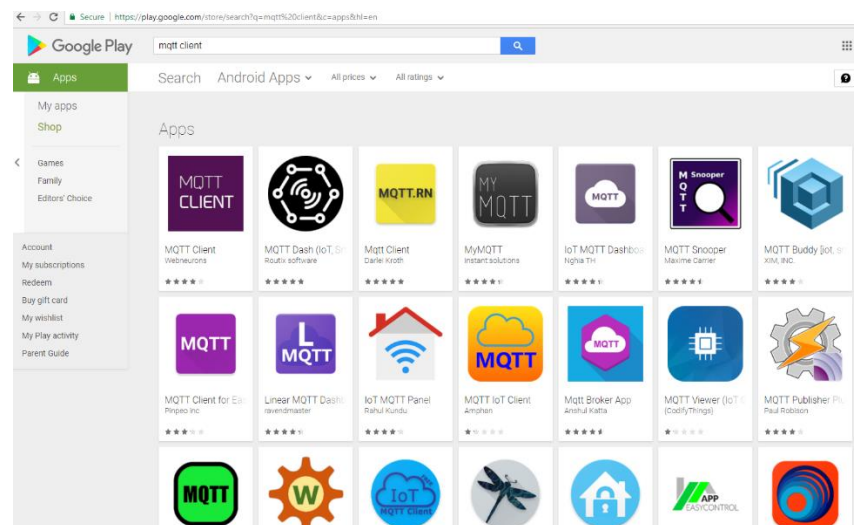


Vaihtoehtoisesti voimme asentaa MQTT-sovelluksen Windows-tietokoneeseen. Avaa Microsoft Store ja etsi osastossa app hakusanalla ”mqtt”.

Kuva 2.1.2. Hakutuloksia sanalle ”mqtt” Microsoft Store:ssa / 2018



Kuva 2.1.3 MQTT -sovelluksia Android-käyttöjärjestelmälle etsittynä Google Play:ssä. / 2018



Asenna MQTTBox joko selainlaajennuksena tai Windows-sovelluksena. Sen käyttö on erittäin yksinkertaista.

MQTTBox voidaan myös asentaa .exe-tiedostosta. Tiedosto voidaan ladata osoitteesta

<http://workswithweb.com/mqttbox.html> . Asennukseen ei tarvita administrator-oikeuksia.

Android- ja iOS-käyttöjärjestelmille kehitettyjä mqtt-sovelluksia on paljon Google Play- ja App Store -sovelluskaupoissa. Valinta voidaan tehdä seuraavilla kriteereillä.

Kun avaat mqtt-sovelluksen, tarjoaa sovellus mahdollisuuden määritellä uusi mqtt client.

Tarkista, että valitsemassasi sovelluksessa vähintään voidaan määritellä:

- MQTT Client name
- MQTT Client Id
- Protocol
- Host
- Username
- Password
- Topic
- Quality of Service
- Payload Type
- Payload

Hyvä lisäominaisuus on mahdollisuus määritellä jokin muu portti mqtt-liikenteelle kuin standardin mukainen 1883.

2.2 Liittyminen mosquitto mqtt broker:iin.

Kuva. 2.2.1 MQTTBox ja asetukset uudelle client:lle.

The screenshot shows the MQTTBox application window with the title 'MQTTBox' and a menu bar containing 'Menu', 'Edit', and 'Help'. The main window is titled 'MQTT CLIENT SETTINGS' and has a 'Client Settings Help' link. The settings are organized into several sections:

- MQTT Client Name:** E_IoTTi
- MQTT Client Id:** 095c40c7-e121-4ecb-a7f6-9
- Append timestamp to MQTT client id?:** ☒ Yes
- Broker is MQTT v3.1.1 compliant?:** ☒ Yes
- Protocol:** mqtt / tcp
- Host:** XXX.XXX.XXX.XX
- Clean Session?:** ☒ Yes
- Auto connect on app launch?:** ☐ No
- Username:** Username
- Password:** Password
- Reschedule Pings?:** ☒ Yes
- Queue outgoing QoS zero messages?:** ☒ Yes
- Reconnect Period (milliseconds):** 1000
- Connect Timeout (milliseconds):** 30000
- KeepAlive (seconds):** 10
- Will - Topic:** Will - Topic
- Will - QoS:** 0 - Almost Once
- Will - Retain:** ☐ No
- Will - Payload:** (empty text area)

At the bottom, there are 'Save' and 'Delete' buttons.

Asetuksiksi tallenna kuvan mukaiset tiedot.

MQTT Client Name; Voit luoda oman client-nimen.

MQTT Client Id; Id voidaan luoda automaattisesti. Valitse lisäksi **append time stamp to MQTT client id**. Näin pienennät mahdollisuutta, että yritäisit ottaa yhteyden brokeriin toisen kerran saman nimisellä client:lla.

Host; Tallenna ip-osoite sille mqtt broker:lle, johon olet liittymässä. HAMK tarjoaa koulutusta ja tutkimusta varten broker:in osoitteessa `iot.xxxxxxx.hamk.fi`. (Saat osoitteen harjoituksen vetäjältä). Mosquito project tarjoaa mosquito – brokerin testitarkoituksiin, lisätietoja: <https://mosquitto.org/> and <https://test.mosquitto.org/>. Käytä mosquitto brokeria, jos muita ohjeita ei annettu.

MQTTBox:ssa ei ole mahdollisuutta valita portin numeroa. MQTT -standardin mukainen portti 1883 on käytössä.

Protocol; valitse mqtt/tcp.

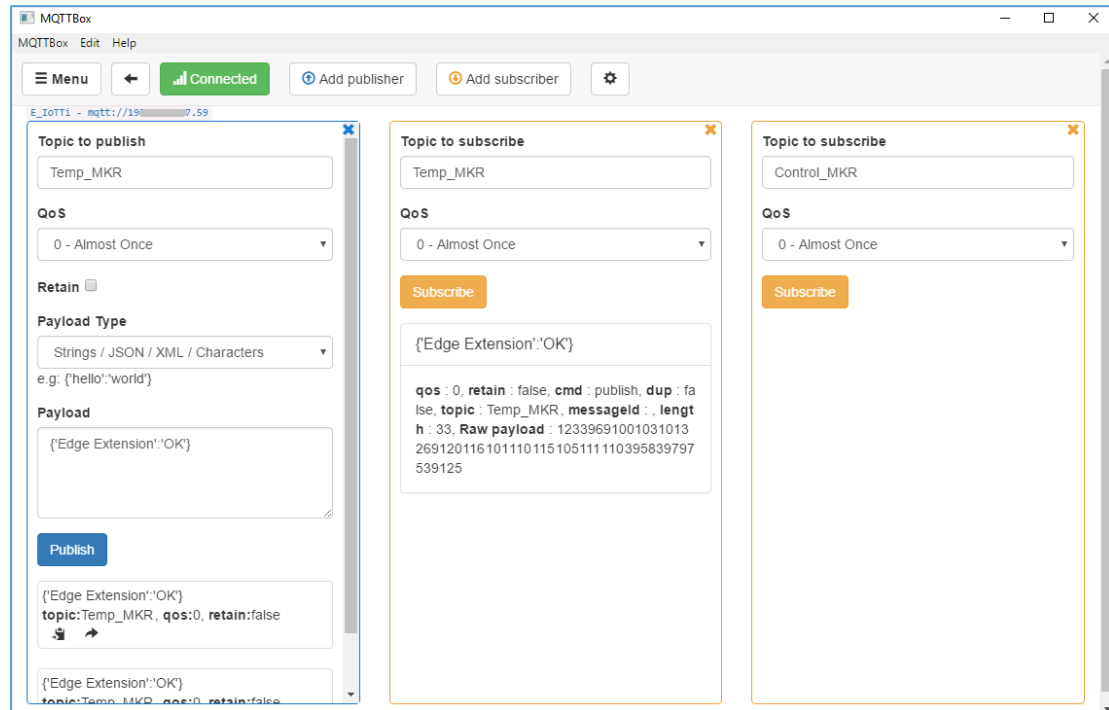
User name, password; Voidaan jättää tyhjäksi silloin, kun MQTT Broker:iin ei ole asetettu käyttäjän tunnistusta.

Yksityiskohtaisia tietoja muista MQTT Broker-asetuksista on verkkosivulla

<https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment>

Kun yhteys on muodostunut, voidaan sen toimintaa testata. Onnistuaksesi yhteyden muodostamisessa on sinun oltava verkossa, jossa liikenne porttiin ja portista 1883 on sallittu. HAMK:n tiloissa tällainen verkko on HAMKvisitor WLAN -verkko. Tyypillisesti tarvittava portti on käytettävissä kotiverkoissa ja matkapuhelimesta jaetussa verkossa.

Kuva 2.2.2 Broker koekäytettynä Windows-sovelluksella MQTTBox. Selainlaajennuksena toteutetussa MQTTBox:ssa käyttöliittymä on täsmälleen samalainen

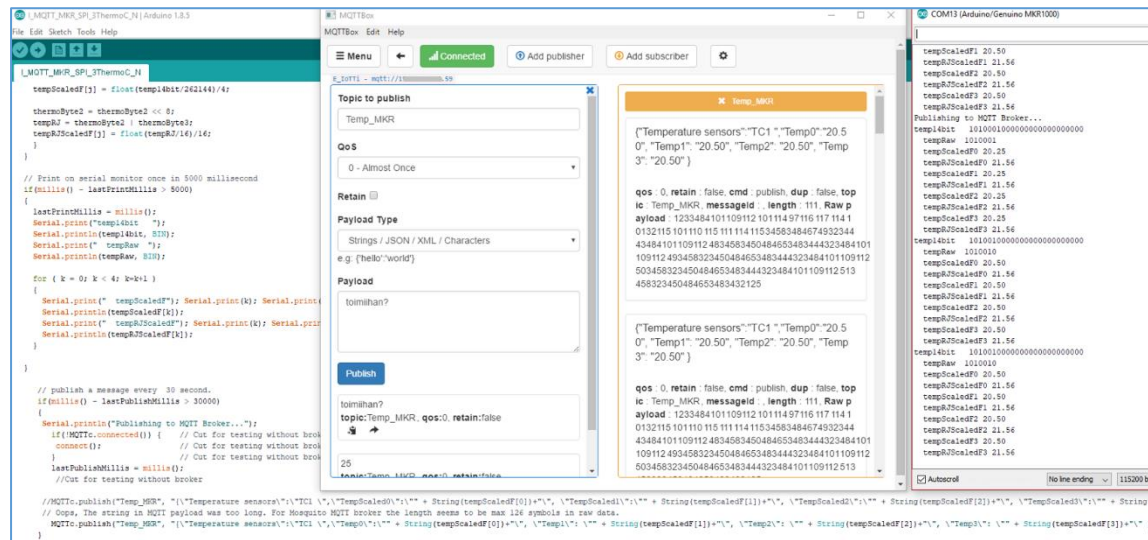


Broker:ia koekäytetään lähettämällä JSON-muotoinen sisältö – payload – uudella aiheella – topic. Kuvassa yllä nähdään mqtt broker toiminnassa.

Broker voidaan koekäyttää myös lähettämällä sanoma todelliselta laitteelta. Tämä ei nyt ole välttämätöntä, jos laitetta ei ole käytettävissä. Esitetään esimerkki käyttäen Arduino-mikro-ohjainkorttia.

Arduino MKR1000 kytketään jännitesyöttöön. Ohjelmanana on edellisessä käyttöönotto-ohjeessa nähty koodi. Laite alkaa lähettää lämpötilatietoja. MQTT Broker:ssa luodaan uusi client ja määritetään se kuuntelemaan aihetta – topic - Temp_MKR. MKR1000 julkaisee – publish – sanomia JSON-muodossa. Kuvassa nähdään, että testi on onnistunut.

kuva 2.2.3 Julkaistaan – publish - useita lämpötila-anturin mittaustuloksia aiheen – topic – Temp_MKR alla. Kirjaututaan tilaamaan – subscribe – juuri samaa aiheetta.



MQTT payload:n suurin pituus näyttäisi olevan 126 symbolia raw-muotoista merkkijonoa. Muutoin subscribe-toiminnolla ei kyetä lukemaan sanomaa. On aika vaikeaa laskea string-muotoisesta merkkijonosta, täsmälleen kuinka monta tavua dataa tämä sisältää. Kuitenkin MQTT Payload:n suurin sallittu pituus pitäisi olla 256 megatavua !

JSON-muoto ei ole välttämätön. Teksti, kokonaisluku, desimaaliluku käyvät myös payload:ksi. MQTT -sanomassa payload lähetetään character string -merkkijonona.

TEHTÄVÄ A.

Tutki MQTT -dokumentaatiosta, mitä sanotaan MQTT topic:sta ja topic-hierarkiasta. Määritä itse useita topic:eja niin, että niillä on jonkinlainen hierarkia. Kirjoita topic:eja vastauksena palautettavaan tekstitiedostoon.

Julkaise sanomia MQTT broker:lle käyttäen määrittämäsi hierarkian alinta tasoa.

Kirjautu tilaamaan MQTT broker:lta äsken määrittämiäsi topic:eja. Kokeile toimintaa käyttäen eri hierarkiatason topic:eja. Mitä tapahtuu wild card:lla # ja + ?

Raporttiin; Kirjoita lyhyt selostus siitä, mitä opit! Kerro erityisesti, miten toimivat + ja # ! Liitä raporttiin ruutukaappaus onnistuneesta kokeilusta.

3. Liittyminen IBM Cloud Watson IoT platform -palveluun

MQTT client -sovelluksessa pyri perustamaan uusi client juuri niillä tiedoilla, joilla perustit laitteen IBM Watson IoT-palvelussa.

Tässä annetaan ohjeet client -sovellukselle MQTTBox. Samat ohjeet soveltuvat muillekin mqtt client -sovelluksille.

Täytä juuri ne tunnistautumistiedot, jotka sait perustaessasi laitteen IBM Watson IoT:ssä:

Server:	<OrganizationID>.messaging.internetofthings.ibmcloud.com
Port:	1883
ClientID:	d:<OrganizationID>:<deviceType>:<DeviceID>
Username:	use-token-auth
Password:	<AuthenticationToken>
Clean Session:	OFF tai ON
SSL:	OFF (ja Watson IoT:ssä muista valita Security Settings, TSL Optional)

Sanoman topic on oltava juuri täsmälleen tietyn muotoinen:

iot-2/evt/<topic>/fmt/json

Merkit <topic> voit korvata lähes millä hyvänsä merkkijonolla. Vältä skandeja ä, ö, .. ja välilyöntiä.

Yllä oleva topic määrittelee, että sanoman payload on oltava JSON-muotoinen. Esimerkki:

```
{ "name": "Pekka", "age": 41, "city": "Oulu" }
```

MQTTBox -sovelluksen käyttö liityttäessä palveluun Watson IoT

MQTT -protokollan mukaista client -sovellusta voidaan käyttää liityttäessä IBM Watson IoT -palveluun. Client-asetukset ja topic-asetukset on kirjattava juuri oikeanlaisina.

Kuva 3.2.1 Asetukset MQTT Client -ikkunassa sovelluksessa MQTTBox. Huomaa valita "No" Append timestamp to MQTT client id !

MQTT Client Name: A_Watson_v8nnas

MQTT Client Id: d-v8nnas-A_DeskTopEditor:EE00

Append timestamp to MQTT client id? ☐ No

Broker is MQTT v3.1.1 compliant? ☒ Yes

Protocol: mqtt / tcp

Host: v8nnas.messaging.internetofthings.ibmcloud.com

Clean Session? ☒ Yes

Auto connect on app launch? ☒ Yes

Username: use-token-auth

Password:

Reschedule Pings? ☒ Yes

Queue outgoing QoS zero messages? ☒ Yes

Reconnect Period (milliseconds): 1000

Connect Timeout (milliseconds): 30000

KeepAlive (seconds): 10

Will - Topic: Will - Topic

Will - QoS: 0 - Almost Once

Will - Retain: ☐ No

Will - Payload:

Save Delete

Please write "use-token-auth" mqtt/tcp or mqtt/tls Just any name

Please write here your authentication token given for this Device ID

Please write "Your organization ID.messaging.internetofthings.ibmcloud"

Please write "d:Your Watson organization ID,Your Watson Device Type, Your Watson Device ID"

Voit myös poistaa asetuksen Auto connect on app launch. Näin voit avata topic-payload-ikkunan yrittämättä samalla yhdistää brokeriin.

Kuva 3.2.2 Topic-määrittely sovelluksessa MQTTBox.

MQTTBox Edit Help

Menu Connected Add publisher Add subscriber

A_Watson_v8nnas - mqtt://v8nnas.messaging.internetofthings.ibmcloud.com

Topic to publish: iot-2/evt/DeskTop1/fmt/json

QoS: 0 - Almost Once

Retain: ☐

Payload Type: Strings / JSON / XML / Characters

e.g: {"hello":"world"}

Payload: {"VariableIntroEE00":"Vuosiluku","VariableValueEE00":2018}

Publish

Topic to subscribe: Topic to subscribe

QoS: 0 - Almost Once

Subscribe

Please write "iot-2/evt/[any name]/fmt/json"

Please select this payload type

Please write in json format

Published message: {"VariableIntroEE00":"Vuosiluku","VariableValueEE00":2018} topic:iot-2/evt/DeskTop1/fmt/json, qos:0, retain:false

Huomaa ettet IBM Cloud Watson:ssa (tai muussakaan kaupallisessa IoT-alustassa) voi lukea subscribe-toiminnolla event-tyyppisiä sanomia. Event on viesti laitteelta IoT-alustalle. Voisit laitteesi oikeilla tunnistautumistiedoilla lukea cammand-tyyppisen viestin subscribe-toiminnolla, jos IoT-alusta julkaisisi publish-toiminnolla komennon eli command-tyyppisen sanoman.

TEHTÄVÄ B

Kokeile erityyppisen tiedon tai mittausarvojen julkaisemista IBM Watson IoT:ssa. Onnistuitko määrittämään topic:n ja julkaisemaan siihen JSON-muotoisen payload:in? Osaatko julkaista tekstiä JSON-muotoisessa payload:ssa? Pystytkö julkaisemaan numeroarvoja eri muodoissa (kokonaisluku, desimaaliluku,...) ?

MQTTBox:lla tai lähes millä hyvänsä mqtt client:lla voit kokeilla julkaista (publish) seuraavanlaisen sisällön (payload). Tämä on tapahtuma (event), jonka lähettää anturilaite.

```
{"d":{"tempCount":156,"temp4position":"Lab1","temp4value":23,"temp5position":"Cellar2","temp5value":18}}
```

Sisältö (payload) on JSON -muotoinen.

Koita julkaista tämä tai jotain vastaavaa IBM Watson:ssa.

Jos työskentelet mikro-ohjainten kanssa, voit koittaa julkaista saman sisällön mikro-ohjaimesta. C++ -kielellä Mbed OS käyttöjärjestelmässä tämä näyttäisi seuraavalta:

```
sprintf(buffer,  
"{\"d\":{\"tempCount\":%d,\"temp4position\":\"Lab1\",\"temp4value\":23,\"temp5position\":\"Cellar2\",\"temp5value\":%d}}\",temp4ADC,temp5ADC);
```

Tämä rivi loisi string -tyyppisen muuttujan buffer. Tämä string lähetettäisiin sisältönä (payload).

Mene omaan IBM Cloud IoT Platform:iin ja avaa oma laite, Device. Katso Recent events, State, Logs.

Saat JSON -objektin. Avaa objekti. Näet property – value -parit, jotka kirjoitit payload:iin.

IBM Cloud Watson IoT JSON -formaatissa {"d":{"hour":7}} avautuisi objektina oikein. Mutta {"d":{"hour":07}} tulostaisi Watson IoT:ssä jotain aivan muuta.... kokeile tätä! Mitä sait? Olet saanut ASCII-symboleita. Jokainen payload-tekstin symboli annetaan ascii-merkkitaulukon mukaisena numerona.

Raporttiin; Kirjoita lyhyt selostus siitä, mitä opit!