

DrvMQTT

DrvMQTT är en drivrutin för MQTT till Web Port.

För att använda DrvMQTT skapas som används för att kommunicera via MQTT.

Installation

Installationen av DrvMQTT görs genom att välja DrvMQTT vid installationen av Web Port. Se *kapitel 1* för mer information kring installationsprocessen.

IO-enhet

För att använda DrvMQTT skapas för en IO-enhet enligt kapitel 3.2.1 i manualen.

Utöver generella inställningar (se kapitel 3.2.4 i manualen) finns följande inställningar för en IO-enhet av typen DrvMQTT.

----- Exempel på inställningar för drivrutin -----

Serveradress

Anger adress till MQTT-brokern

Ex. mqttserver.host.com

Port

Anger port till MQTT-brokern

Kryptering

Val av krypteringstyp

Autentisering

Certifikat för både klient och server

Eller användarnamn och lösenord

Används i kombination med autentiseringsuppgifter.

Autentiseringsuppgifter

Anges beroende på val som gjort för autentisering.

Vid certifikat är formatet följande:

Klientcertifikat:Servercertifikat

Där Klientcertifikat resp. Servercertifikat är CommonName på certifikaten som installerats i lokal certstore för local machine-

Vid användarnamn gäller formatet:

Användarnamn:lösenord

Klient-Id

Ger möjlighet att ange eget Klient-Id (Provider Id). Om det lämnas tomt slumpas ett Id.

Taggar

För mer information om hur tagglistor och taggar skapas se kapitel 4 i manualen.

" Taggadresser:

Adressen anger den topic som taggen skall prenumerera på. Ex. sensor/ fastigheter /+ /+ /payload

Svaret kommer ofta komma som en sträng i JSON-format då finns det två olika möjligheter att få ut de värden som önskas.

Ett alternativ är att ange ett JSON-filter enligt format: TOPIC|JSON-filter

Svaret som kommer från brokern filtreras därefter med ett JSON-filter för att hämta ut ett enskilt värde.

De båda delarna separeras av "pipe"-tecken.

För att görs booleska jämförelser skrivs dess efter adressen inom {}. (exempel {>0})

Ett annat alternativ är att skicka datan till ett anropat script. Detta är lämpligt att använda då den data man söker inte kommer i klartext utan behöver behandlas för att t.ex veta vilken tagg den skall skrivas till eller för att få ut t.ex. mätvärdet. Exempel nedan.

Om man önskar sätta upp en topic dit man endast vill skriva och ej prenumerera på meddelanden, anges ! före adressen. T.ex. !/sensor/xxx/payload

" Exempel:

JSON-filtret använder Newtonsoft JSONs SelectToken-funktion.

Exempel på hur den kan användas:

JSON-svar

```
{
  "timeSeries": [{
    "validTime": "2015-10-12T04:00:00Z",
    "parameters": [{
      "name": "msl",
      "levelType": "hmsl",
      "level": 0,
      "values": [
        1031
      ]
    }
  ]
}]
```

```
    }}  
}
```

Välja validTime: timeSeries[0].validTime

Välja värdet (1031) i parameter med namn msl:
timeSeries[0].parameters[?(@.Name == 'msl')].values[0]

I vissa fall kommer datan som som en array utan namn (börjar/slutar på []) och då kan filtret t.ex. börja på [0].validTime

Mer information hittas på newtonsofts hemsida:
<https://www.newtonsoft.com/json/help/html/SelectToken.htm>

" Exempel:

Mätdata vi vill ha kommer som ett encoded payload som behöver konverteras för att bli läsbar.

```
{ "devEui": "94193a010600010d", "sensorType": "other", "timestamp": "2019-11-05T19:00:37.07Z",  
  "payload": "000265830801fb1b1f", "spreadingFactor": "7", "rssi": "-111", "snr": "7",  
  "gatewayIdentifier": "246", "fPort": "2" }
```

Vi väljer här att skicka hela paket vidare till ett script som avkodar payloaden
Som adress anges: sensor/fastigheter/+/+/payload{script#blinkCMA11L(jsondata=DEFAULT)}
blinkCMA11L är scriptet som anropas till scriptet skickas hela paketet som variabeln jsondata.

Exempel på script:

```
// Då data i detta fall kom i en array väljer vi ut första elementet i arrayen  
var jd = jsondata[System.Int32.Parse(0)];  
  
// Vi tar sedan ut vår payload  
var payload = jd["payload"];  
  
//I detta fall behöver vi göra en operation där vi tar vi tar plockar ut och byter plats på hextal samt  
omvandlar dessa till flyttal.  
  
var temperature =  
parseFloat(System.Int32.Parse(payload.ToString().Substring(8,2)+" "+payload.ToString().Substring(6,2),  
System.Globalization.NumberStyles.HexNumber));  
  
//Vi avslutar med att returnera temperaturen som skrivs till taggen  
return temperature;
```