

**Studenten:**

Arshiya Sharifi 2139515 [a.sharifi@student.avans.nl](mailto:a.sharifi@student.avans.nl)

Nicholas Le Roux 2156763 [nkw.leroux@student.avans.nl](mailto:nkw.leroux@student.avans.nl)

Robin Koedood 2151018 [jj.koedood1@student.avans.nl](mailto:jj.koedood1@student.avans.nl)

Timo Jenkins 2153042 [tjr.jenkins@student.avans.nl](mailto:tjr.jenkins@student.avans.nl)

**Organisatie:** Avans Hogeschool

**Plaats, datum:** Breda, 13-04-2022

**Versie:** 2.0

**MQTT handleiding**

TI-EMS Energie Management Systems

**MQTT OpenEMS**

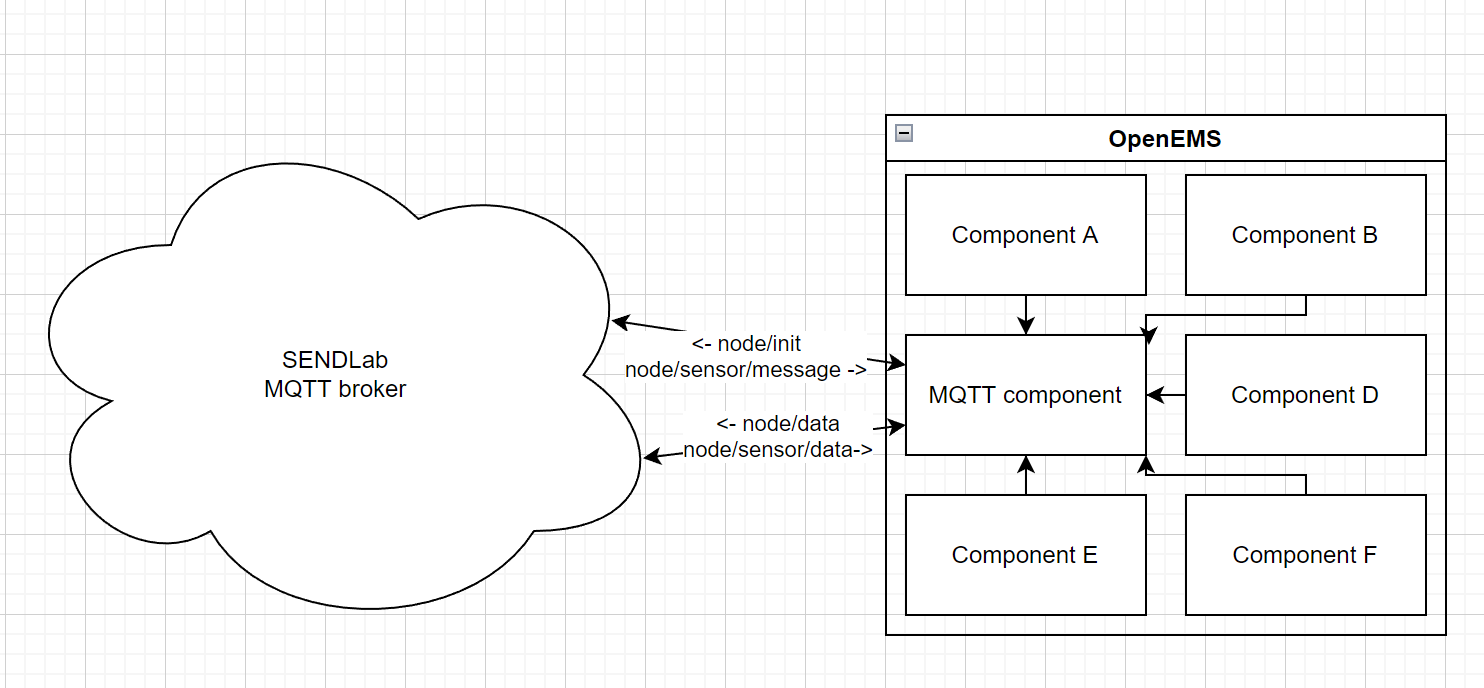
OpenEMS is een opensource project en bevat veel componenten die klaar zijn voor gebruik. Een van die componenten is voor MQTT. Mijn taak was de componenten verder uit te breiden en vervolgens OpenEMS met het Smart Netwerk te koppelen. Het is mij gelukt om data van OpenEMS naar de MQTT-broker van het SENDLab te sturen. Het is ook gelukt om data van de MQTT-broker te lezen door op topics te subscriben. Op de Smartnetwerk zijn verschillende nodes beschikbaar. Die zijn verschillende soorten meters met eigen data. Je kan vanuit de component uit verschillende node types kiezen. De standard optie print alleen de opgehaalde data in de console. Een van de nodes is voor een Smartmeter, een zelf ontwikkelde meter van het lectoraat Smart Energy. Ik heb hiervoor een Smartmeter component geschreven om de data van de broker te parsen. De data die van de subscribed topic ophaalt wordt dan geparsed en doorgestroomd naar de Smartmeter component. De component communiceert met de edge UI en toont de opgehaald data op.

De component heeft nog bugs en is nog niet klaar voor productie. Wat nog moeten worden gedaan is de dataverwerking van wat de client (MQTT-component) ophaalt/ontvangt van de broker. Daar zijn meerdere nodes waarvoor elke een component moet voor worden geschreven om bruikbaar te zijn in OpenEMS. De data wordt nog niet gebruikt om een component van te maken en de opgehaalde data is niet toonbaar in de ui.

De volgende stap zou zijn om meer functionaliteit toe te voegen in de Smartnetwerk zelf. Het is nu alleen mogelijk data naar toe te schrijven en data van te lezen. Het is nog niet mogelijk om commandos vanaf de broker uit te voeren via een topic of de topics te beveiligen. Hiervoor moet dan ook de functionaliteit worden geïmplementeerd in de MQTT custom component. De enige component die is gemaakt voor de Smartnetwerk is de Smartmeter. Als je andere nodes wil gebruiken vanaf de Smartnetwerk moet je een nieuwe component voor worden geschreven.

Om gebruik te maken van andere nodes vanaf de Smartnetwerk, dan moet een nieuwe component er voor worden geschreven.

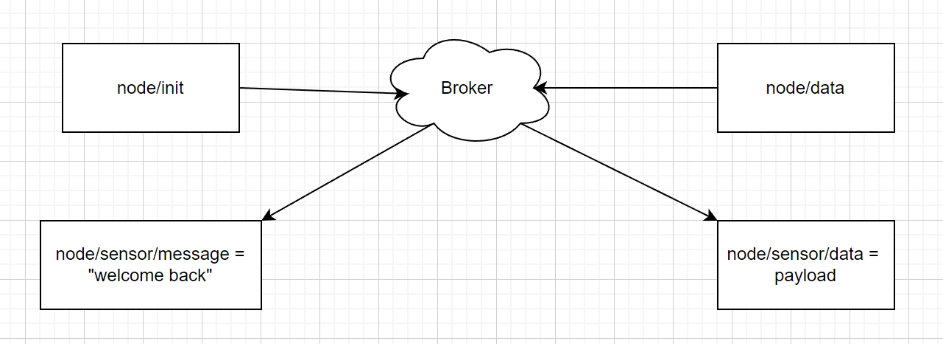
**Het structuur en communicatie van het Smart Netwerk met OpenEMS**



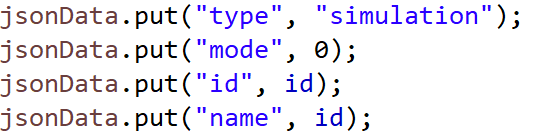
MQTT-component haalt alle data op van componenten A-F na 15 sec opstart. Stuurt naar MQTT-broker via node/init. Component ontvangt node/sensor/message terug. MQTT-component stuurt dan elke 1~5 seconden alle data van OpenEMS naar de MQTT-broker via node/data. Component ontvangt de data.

**Hoe communicatie werkt op het Smart Netwerk**

De variable “sensor” van node/sensor/... of id is een variable die verschilt per node. De variable kan worden gezet in het init bericht.

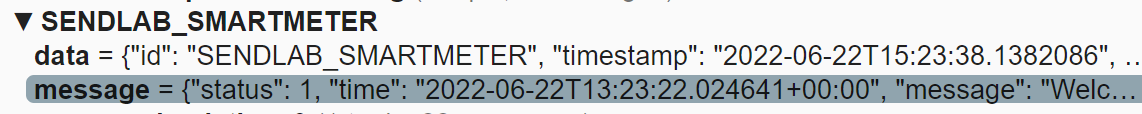


Code snippet: set some of node/init parameters.

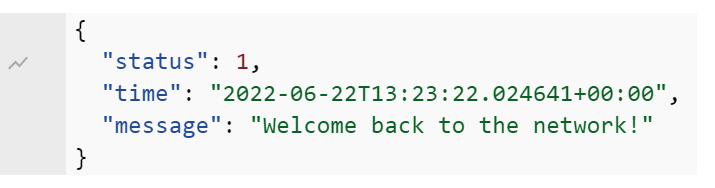


Init moet ook de measurements en/of actuatoren bevatten. De measurements zijn de actuele data die op het Smart Network wordt opgeslagen. Het bestaat uit een array van measurements. Een measurement bestaat uit: een name, een description, en een unit (bijv Celsius, kW/h).

Voorbeeld van node/init en node/data server responses.



**Node/init**

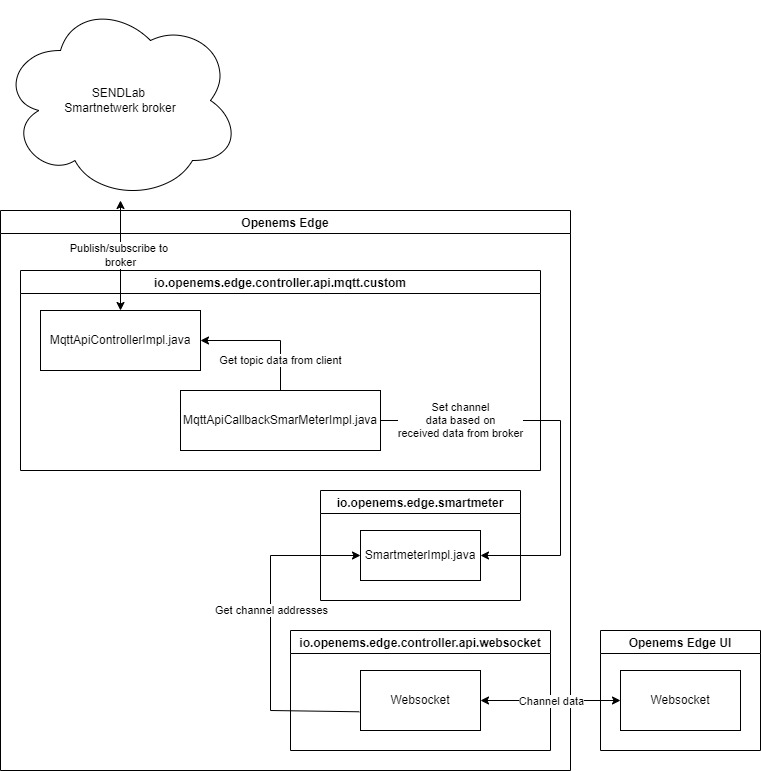


**Node/data**



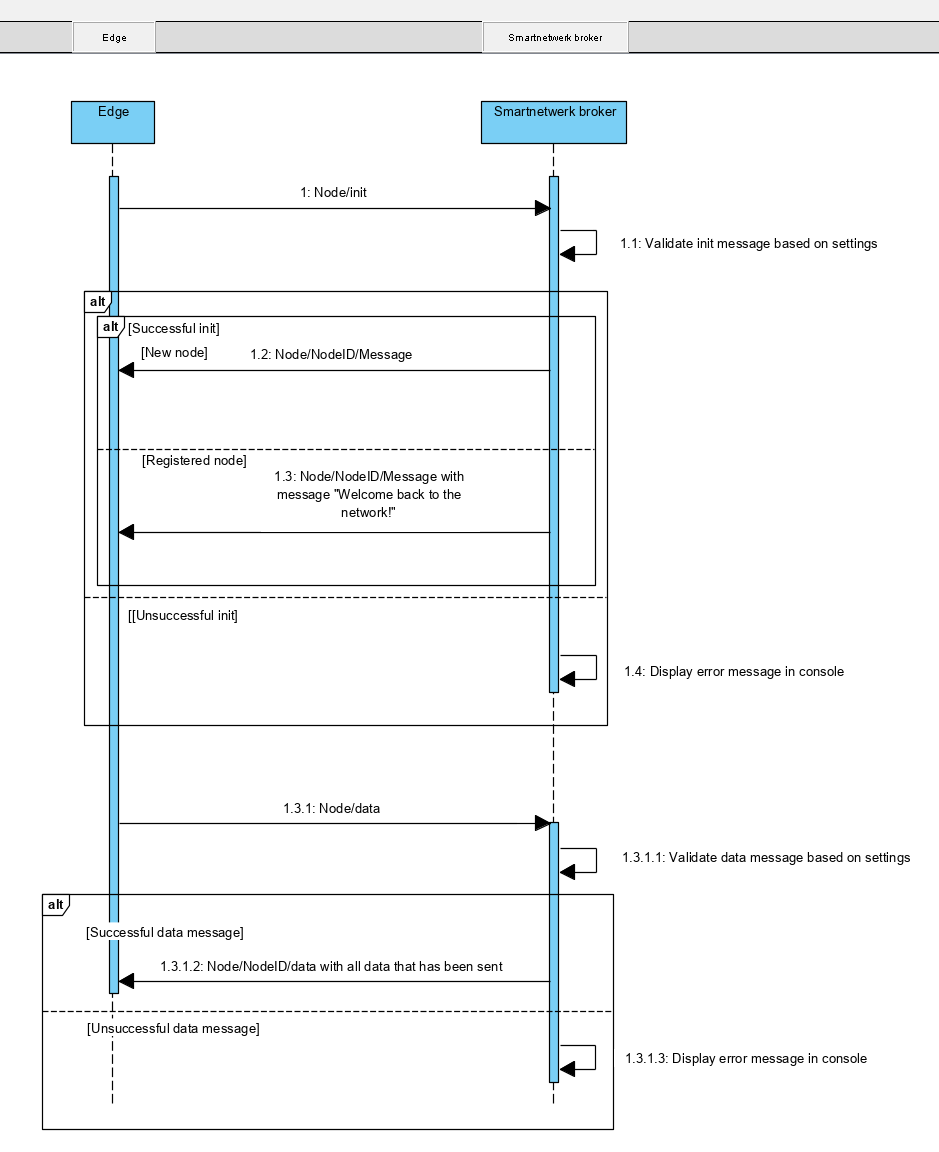
**Hoe je data kan ophalen vanaf de mqtt broker**

Elke MQTT broker heeft topics waar berichten kunnen worden verstuurd. Hiervan kan ook de data van worden gelezen via op een topic te subscriben. In het diagram onderaan staat een voorbeeld van hoe het werkt in OpenEMS en het Smartnetwerk.

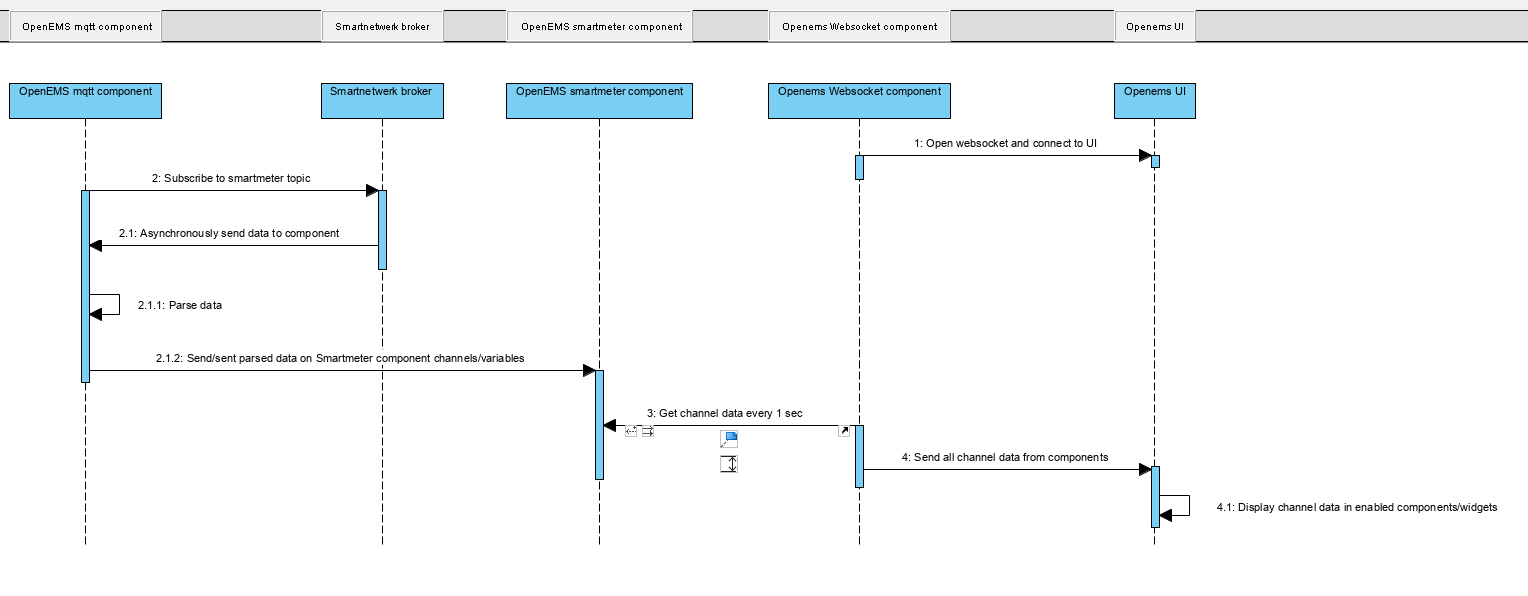
****

De data wordt opgehaald via een callback (mqttapicallbacksmartmeterimpl) die draait asynchronys van de client. Het wacht totdat een bericht binnen komt van de broker. Wanneer een bericht binnen komt wordt het geparsed en omgezet naar bruikbare data. Daarna wordt het verstuurd naar de Smartmeter component. De data wordt dan opgeslagen in de Channels in de Smartmeter. Elke seconde wordt de data channel verstuurd naar de websocket die verbindt de edge met de uit. De data wordt vervolgens dat verstuurd naar de UI en getoond.

Sequence diagram die toont hoe publish werkt tussen OpenEMS en de broker.

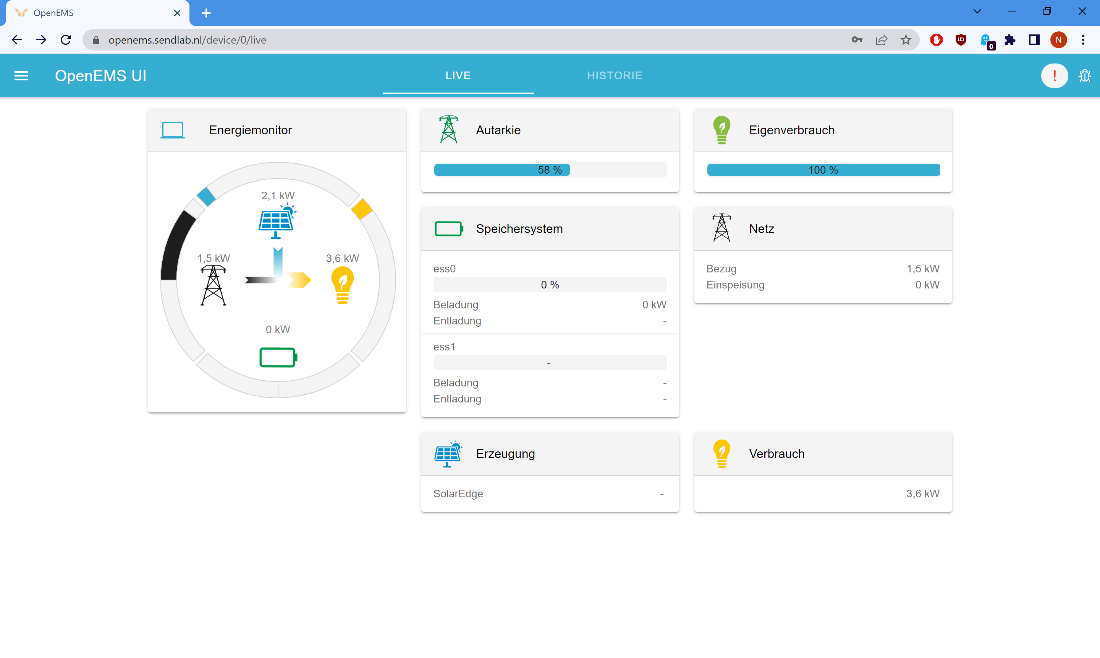


Sequence diagram die toont hoe subscribe werkt met een broker en een smartmeter topic

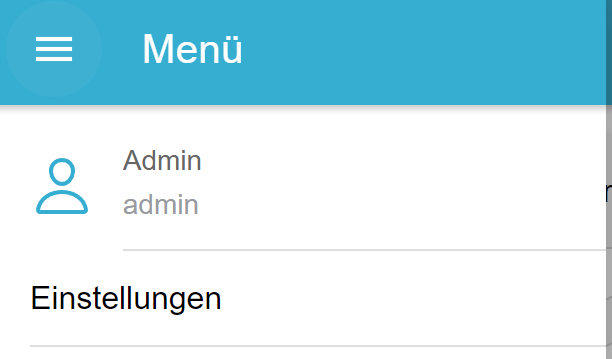


**Nieuwe MQTT-component toevoegen en instellen via UI**

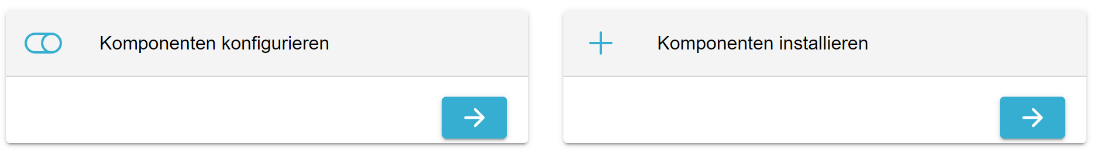
1. Open de ui en login als Admin (https://openems.sendlab.nl/index)



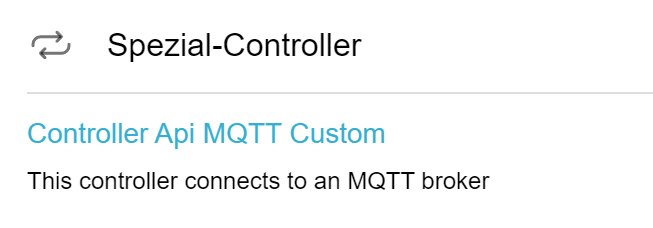
1. Druk op “Einstellungen”



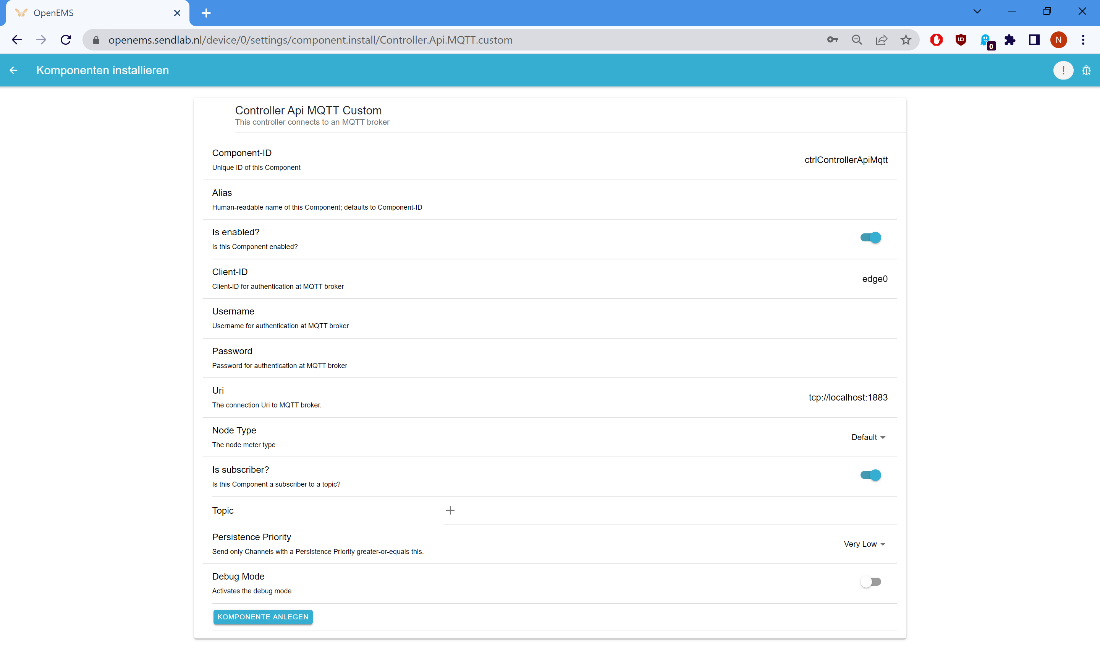
1. Druk op “Komponenten installeren”



1. Onder “Spezial-Controller” staat de custom versie van de MQTT-component



1. Select de Controller APi MQTT Custom component.



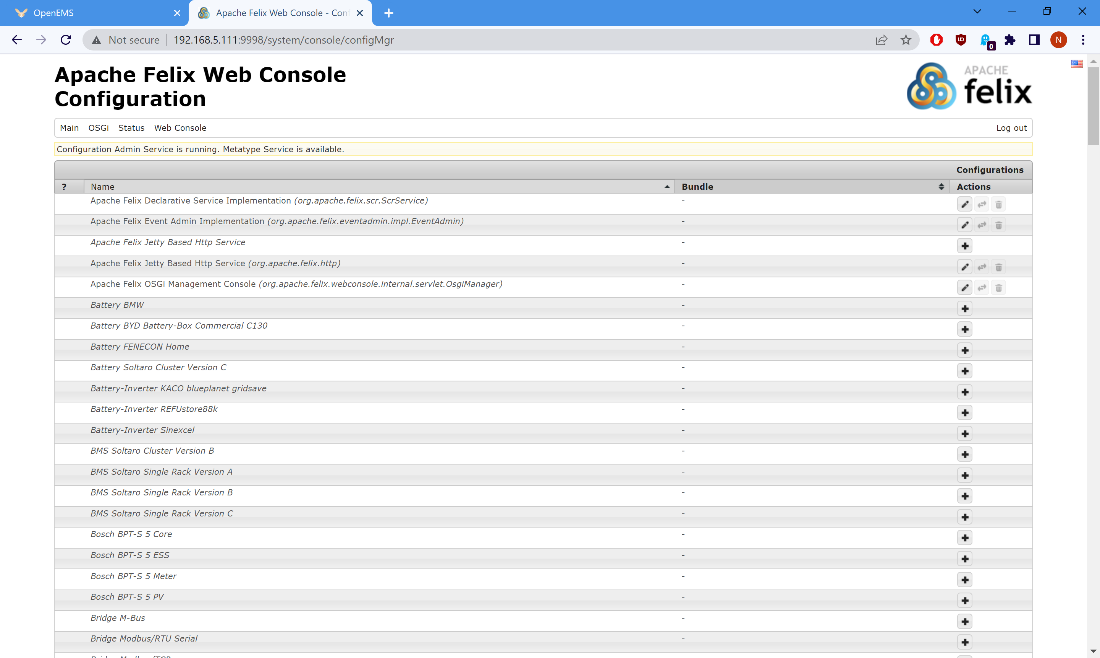
1. Vul in de waardes van de MQTT-broker.

Info voor SENDLab

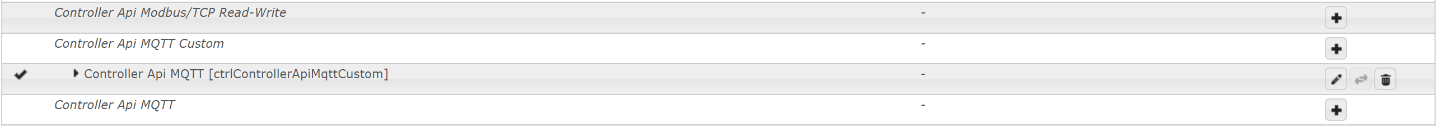
* Username: node
* Password: smartmeternode
* Uri: tcp://sendlab.nl:11884

**Nieuwe MQTT-component toevoegen en instellen via backend**

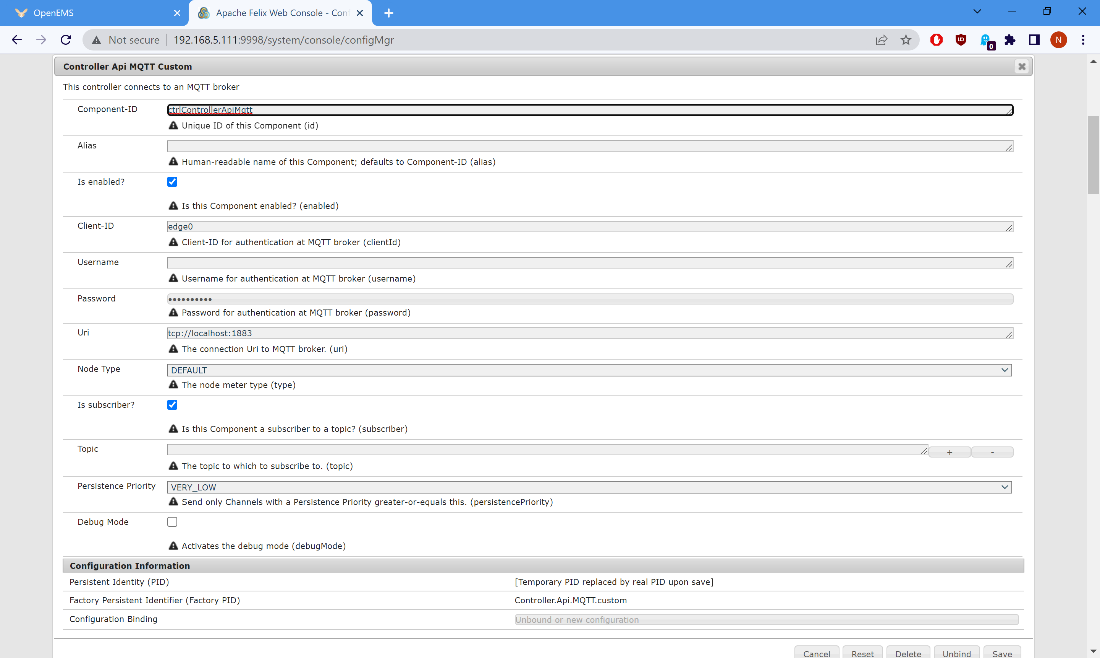
1. Open backend en login. (http://192.168.5.111:9998/system/console/configMgr)



1. Selecteer de MQTT custom component



1. Vul in de gegevens van de MQTT broker.

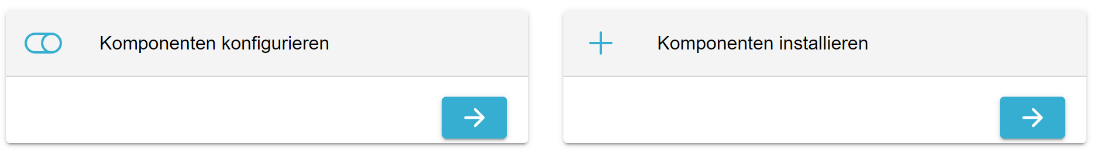


Info voor SENDLab

* Username: node
* Password: smartmeternode
* Uri: tcp://sendlab.nl:11884

**Bestaande MQTT-component instellen**

Volg stap 1 en 2 van “**Nieuwe MQTT-component toevoegen en instellen via UI”**.

1. Druk op “Komponenten konfigurieren”.

Volg de rest van de stappen van “**Nieuwe MQTT-component toevoegen en instellen via UI”**.