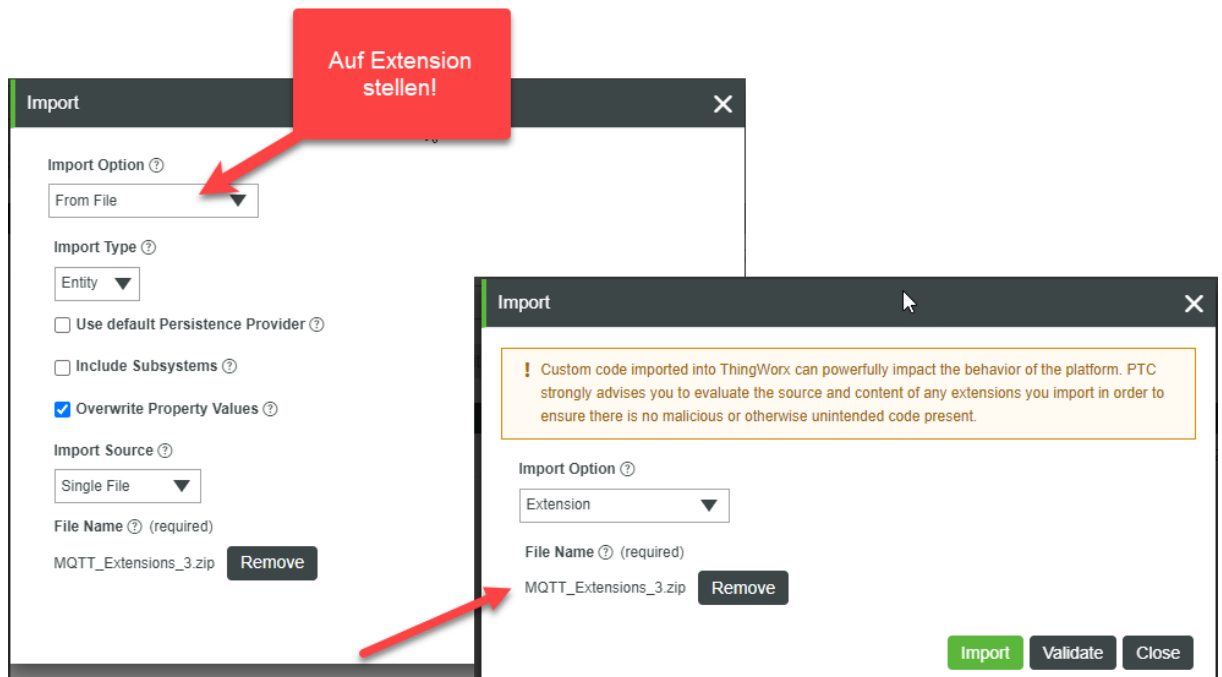
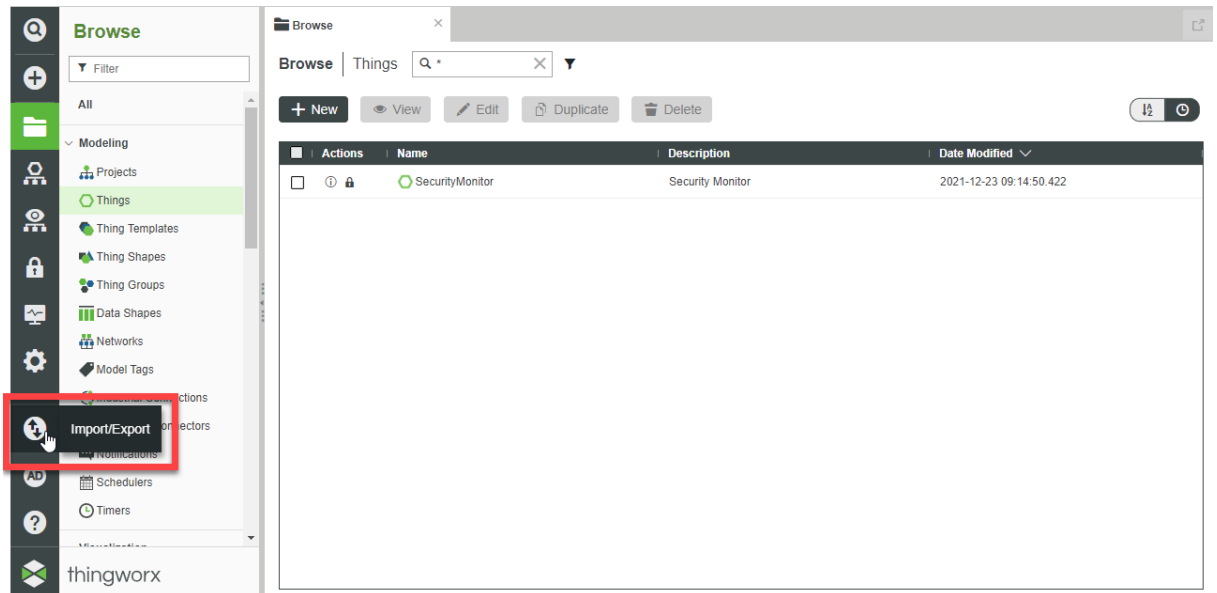


MQTT-VERBINDUNG HERSTELLEN

1. Extension installieren

MQTT_Extensions_3.zip → nicht entpacken!




2. Neues Thing erstellen als Template MQTT wählen

Thing: New Thing - 2 * To Do (1) Save Cancel

General Information Properties and Alerts Services Events

General Information

 **Name** (required)
22L01_SUM_Phone

Description

Advanced Search

- New Thing Template
- MQTT**
- MQTTConnection
- MQTTSubscriber

Value Stream

Search Value Streams

3. Permissions setzen

Thing: 22L01_SUM_Phone To Do Save Cancel More

General Information Properties and Alerts Services Events Subscriptions Configuration Permissions

Configuration

Permissions 22L01_SUM_Phone Browse

Permissions Entities Save Done Cancel

Visibility Run Time Design Time

Visibility

Remove Org or Org No Organizations

Search Organizations

- Composer
- Development
- es-authorization-org
- es-public-access-org
- Everyone**
- SolutionCentral

Run Time

All Properties, Services, and Events

Remove Bulk Set Search Users or Groups

User or Group

No Users or User Groups

Property, Service, or Event Overview

Property Read **Property Write** **Service Execute** **Event Execute** **Event Subscribe**

User or Group	Property Read	Property Write	Service Execute	Event Execute	Event Subscribe
System	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

4. MQTT-Server Daten eintragen

The screenshot shows the configuration page for a device named 'Thing:22L01_SUM_Phone'. At the top, there are tabs for 'General Information', 'Properties and Alerts', 'Services', 'Events', and 'Subscriptions'. The 'General Information' tab is active. Below the tabs, there is a 'Mappings' section with a '+ Add' button. Under 'Mappings', there is an 'Actions' table with one entry: 'subscribe'. Below this, there is a 'JDBC Settings' section. The 'clientIdFormat' field contains the text 'immer_landers'. The 'useSSL' checkbox is unchecked. The 'serverName' field contains 'letto.htlwrn.ac.at'. The 'serverPort' field contains '1883'. The 'userId' field contains 'iot2021'. There are also fields for 'password', 'New Password', and 'Confirm Password', all of which are currently empty. Three red callout boxes with arrows point to specific fields: the first points to 'clientIdFormat' with the text 'Eindeutige Client ID. Darf es pro MQTT-Broker nur einmal geben!'; the second points to 'serverName' with the text 'Alternativ können öffentliche MQTT-Broker verwendet werden'; the third points to 'userId' with the text 'Bei öffentlichen Brokern eventuell leer'.

Thing:22L01_SUM_Phone * ?

To Do Save Cancel More

General Information Properties and Alerts Services Events Subscriptions

Mappings + Add

Actions subscribe

No data

JDBC Settings

clientIdFormat

immer_landers

useSSL

serverName

letto.htlwrn.ac.at

serverPort

1883

userId

iot2021

password

New Password

Confirm Password

Eindeutige Client ID
Darf es pro MQTT-Broker
nur einmal geben!

Alternativ können öffentliche
MQTT-Broker verwendet
werden

Bei öffentlichen Brokern
eventuell leer

Der MQTT-Broker in Wiener Neustadt unterstützt auch die Verschlüsselte Verbindung (SSL). In diesem Fall ist als Port 5003 zu verwenden.

Nach dem Speichern sollte das Verbindungssymbol grün werden:



5. Subscriben von Topics

man kann nur Daten empfangen, wenn man Topics subscribed für die man auch die entsprechenden Rechte hat.

Für den User „iot2021“ ist das zum Beispiel edu/iot2021/#

d.h. alle Sub-Topics unterhalb von edu/iot2021/ lassen sich lesen als auch schreiben

Bei öffentlichen Brokern gibt es diese Einschränkungen meistens nicht.

Die Schreib- und Leserechte werden am Server festgelegt und können von „normalen“ Usern nicht geändert werden.

The screenshot shows the ThingsBoard interface for a thing named '22L01_SUM_Phone'. The 'Configuration' tab is selected. A 'Mappings' dialog is open, showing a list of mappings with 'subscribe' checked. The 'name' field is set to 'int_Testvalue' and the 'topic' is 'edu/iot2021/SUM/Testvalue'. A red box labeled 'Gleichnamiges Property erstellen' points to the 'Properties' tab where a new property 'int_Testvalue' is being added.

6. Werte zum Testen via MQTT Explorer senden

The screenshot shows the MQTT Explorer application. On the left, the topic tree is expanded to 'edu/iot2021/SUM'. On the right, the 'Publish' interface is shown with the topic 'edu/iot2021/SUM/Testvalue' selected. The 'raw' radio button is selected, and the value '1' is entered in the input field. The 'PUBLISH' button is highlighted.

7. Werte in TWX aktualisieren:

The screenshot shows the 'Thing:22L01_SUM_Phone' interface. The 'Properties' tab is selected and highlighted with a red box. Below the tab, there are buttons for 'Add', 'Duplicate', 'Delete', 'Manage Bindings', and 'Refresh' (the last one is highlighted with a red arrow). A table lists properties with columns: Name, Actions, Source, Default Value, Value, and Alerts. The 'Value' column for the 'int_Testvalue' property is highlighted with a red box and contains the value '1'. Below the main table, there is a section for 'Inherited Properties' with a 'MQTT' sub-section containing a table with columns: Name, Actions, Source, Default Value, and Value. The 'isConnected' property has a checked checkbox in the 'Value' column. The 'lastConnection' property shows a timestamp '2022-03-11 07:31:27.179'.

MQTT JSON VERARBEITEN

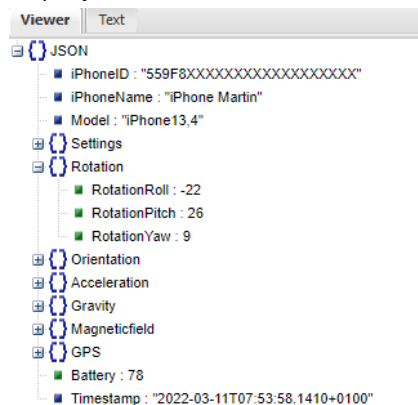
Viele Endgeräte liefern nicht für jeden Datenpunkt ein eigenes Topic sondern senden die Daten im JSON-Format.

Ein Datensatz kann weitere Datensätze enthalten, dadurch sind beliebig tief verschachtelte Strukturen möglich. Als Beispiel sind hier die Daten der iOS App Cedalo angeführt. Diese App sendet unter anderem eine Vielzahl von Sensordaten des iPhones als JSON an einen MQTT-Broker. (Einrichtung der App siehe im Kapitel „MQTT-Datenquellen“)

The screenshot shows the 'MQTT Explorer' interface. On the left, a tree view shows the topic structure: 'letto.htlwn.ac.at' > 'edu' > 'mb' > 'iot2021' > 'SUM' > 'iPhone'. The main panel displays a JSON message received on the topic 'edu/iot2021/SUM/iPhone'. The message is a JSON object with fields like 'Model', 'Settings', 'Interval', 'IntervalOnChange', 'IntervalCycle', 'Threshold', 'ThresholdValue', 'Rotation', 'RotationRoll', and 'RotationPitch'. The 'Publish' section at the bottom shows the topic 'edu/iot2021/SUM/Testvalue' and a 'PUBLISH' button.

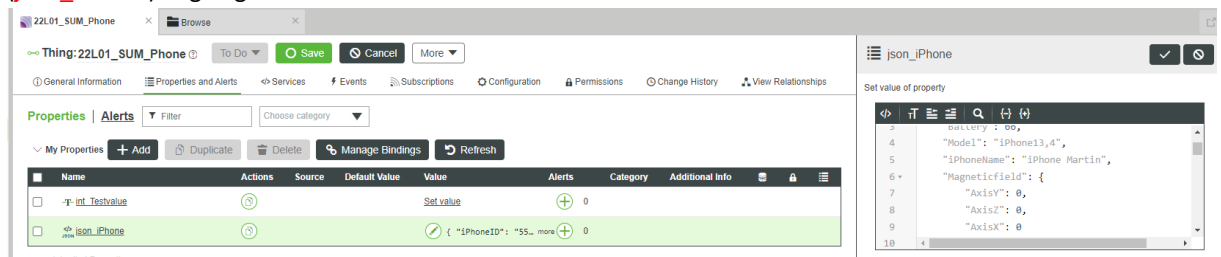
Einerseits können im MQTT-Explorer die Daten auch bereits sehr komfortabel angezeigt werden, weiters gibt es aber auch noch diverse Viewer um die Datenstruktur besser zu visualisieren:

<http://jsonviewer.stack.hu/>



Um die so gewonnen Werte im MQTT-Thing in einzelne Propertys zu kopieren müssen folgende Schritte durchgeführt werden:

1. So wie bei dem ersten Wert wird das Topic wieder subscribed und das entsprechende Property (**json_iPhone**) angelegt.

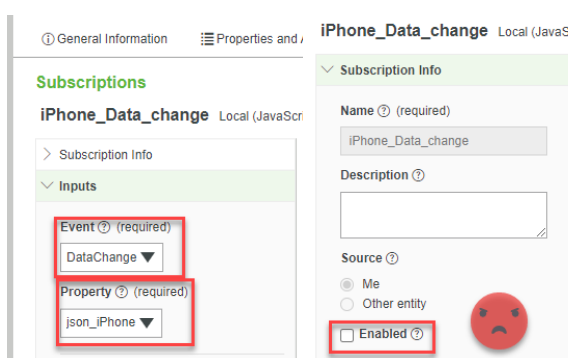


Wenn man den Datentyp JSON angibt, erfolgt die Anzeige in einer besser lesbaren Form.

Weiters legt man noch alle Propertys an, die man explizit „haben“ will – in diesem Beispiel

int_iPhone_Battery

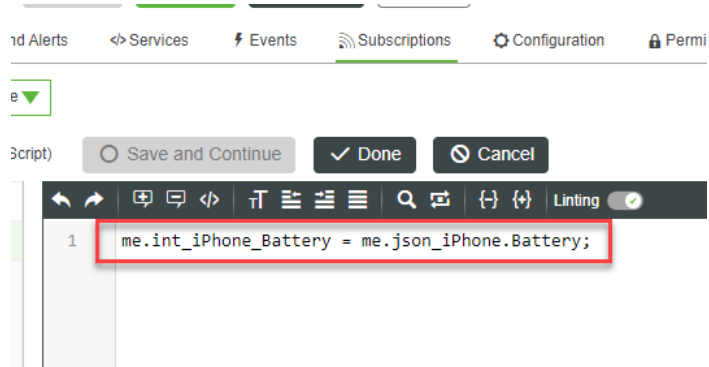
2. Subscription anlegen
(Es wird die Bedingung für die Ausführung des Programmcodes angegeben)



deaktiviert:

!!! ACHTUNG Subscriptions sind standardmäßig

3. Nun werden in die Property für die einzelnen Werte die entsprechenden Werte aus dem JSON kopiert:



Weitere Werte können einfach durch das hinzufügen zusätzlicher Zeilen kopiert werden.

ACHTUNG **Me.** ist nicht **me**.

Eine explizite Fehlermeldung erscheint nicht!

