

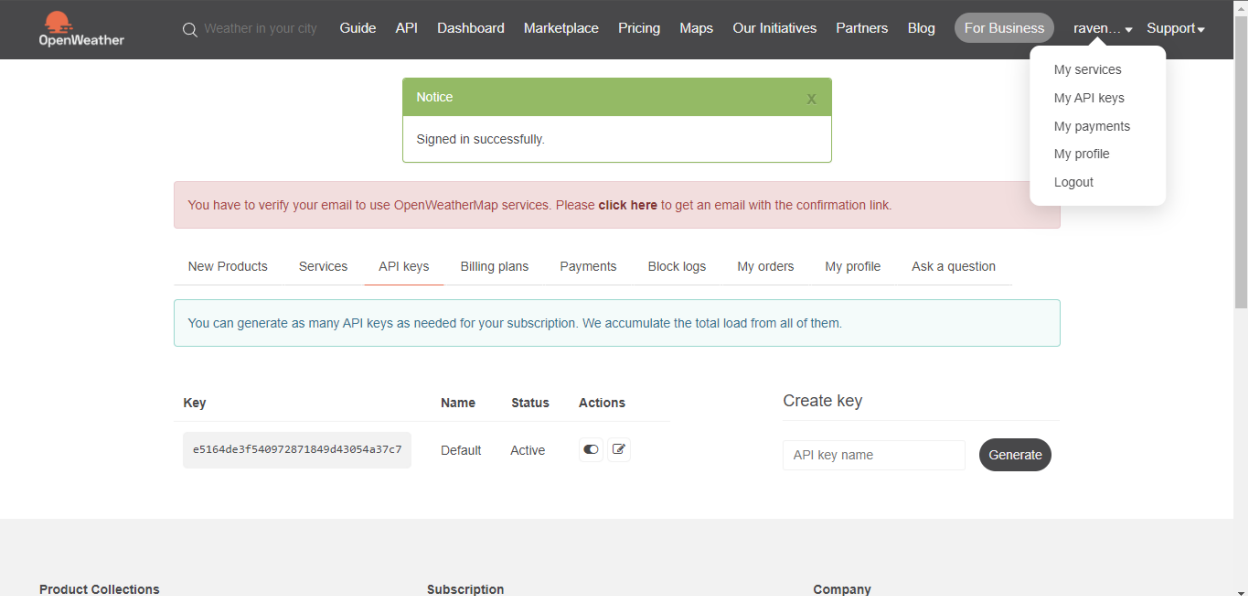
Weather Monitoring System Using ThingsBoard and REST API

Raven Daniel Martin

Dengan memanfaatkan thingsboard, buatlah system pembacaan cuaca dengan pemanggilan REST API.

Kita dapat menggunakan info cuaca untuk logika pemrosesan data tertentu atau hanya untuk melacak riwayat dan mengaktifkan visualisasi info ini di dashboard. Kita akan mengkonfigurasi ThingsBoard Rule Engine untuk secara otomatis mendapatkan informasi cuaca menggunakan REST API.

Langkah pertama adalah mencari web penyedia data cuaca, dalam paper ini kita gunakan web openweathermap.org. Setelah mendaftar di web tersebut, kita beralih ke laman My API keys untuk mendapatkan key API yang nantinya akan digunakan pada ThingsBoard.



The screenshot shows the OpenWeatherMap website's 'API keys' management page. At the top, there's a navigation bar with links like 'Guide', 'API', 'Dashboard', 'Marketplace', 'Pricing', 'Maps', 'Our Initiatives', 'Partners', 'Blog', 'For Business', and 'Support'. A user is logged in as 'raven...'. A green notice box says 'Signed in successfully.' Below this, a red banner states: 'You have to verify your email to use OpenWeatherMap services. Please [click here](#) to get an email with the confirmation link.' The main content area has tabs for 'New Products', 'Services', 'API keys' (selected), 'Billing plans', 'Payments', 'Block logs', 'My orders', 'My profile', and 'Ask a question'. A light blue box informs: 'You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.' Below this is a table with columns 'Key', 'Name', 'Status', and 'Actions'. One key is listed with ID 'e5164de3f540972871849d43054a37c7', name 'Default', and status 'Active'. To the right of the table is a 'Create key' section with an input field for 'API key name' and a 'Generate' button. At the bottom, there are links for 'Product Collections', 'Subscription', and 'Company'.

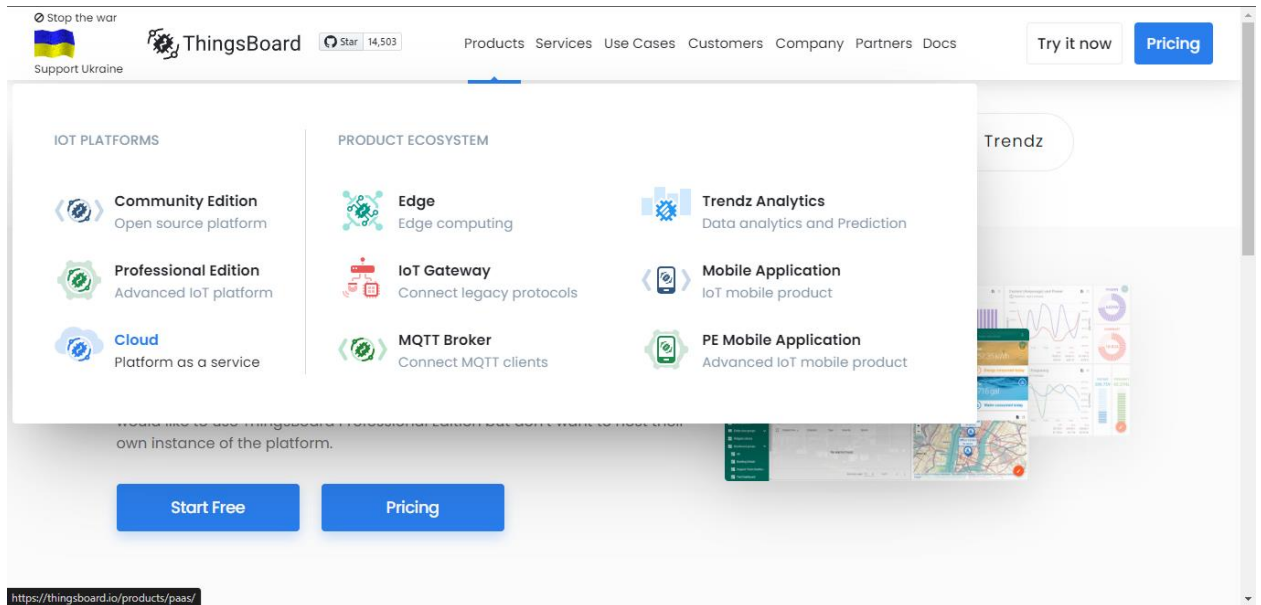
Key	Name	Status	Actions
e5164de3f540972871849d43054a37c7	Default	Active	Toggle Edit

Create key

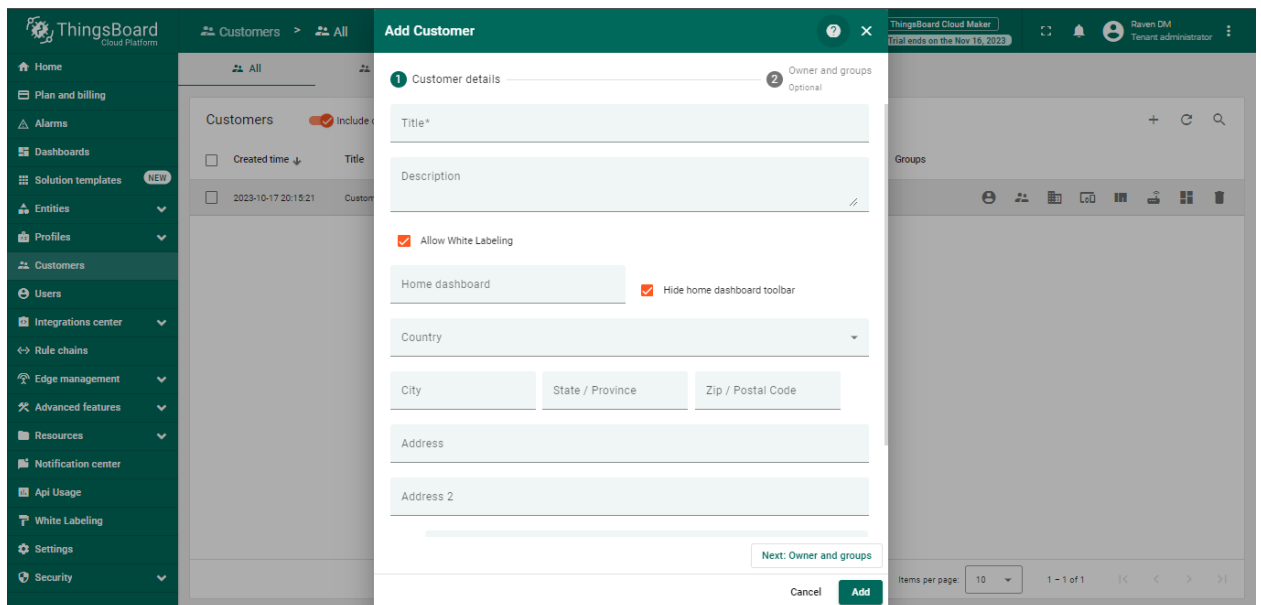
API key name Generate

Langkah selanjutnya adalah masuk ke web ThingsBoard, selanjutnya pilih product, dimana pada paper ini product yang dipilih adalah cloud, kemudian pilih Start Free. Kita akan disambungkan ke laman

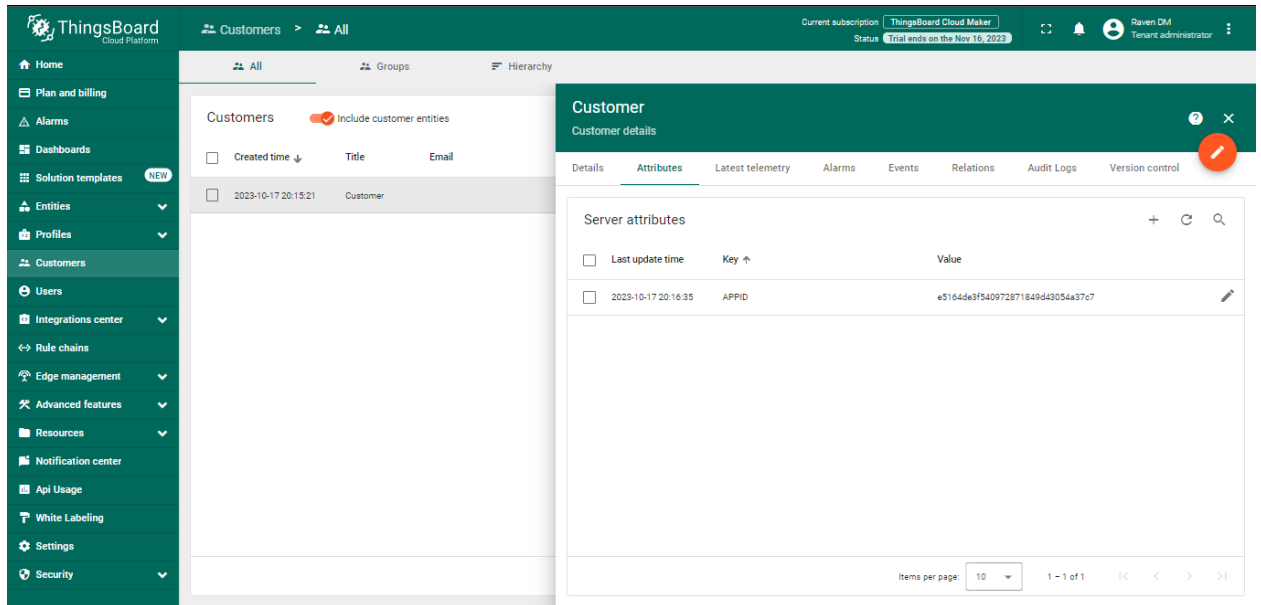
pendaftaran atau dapat melakukan sinkronisasi akun dengan pihak ketiga seperti yang digunakan pada paper ini adalah google account.



Langkah awal pada ThingsBoard adalah menetapkan customer dengan cara pilih Customers -> pilih tanda “+” di pojok kanan dan input title sesuai penamaan yang diinginkan.

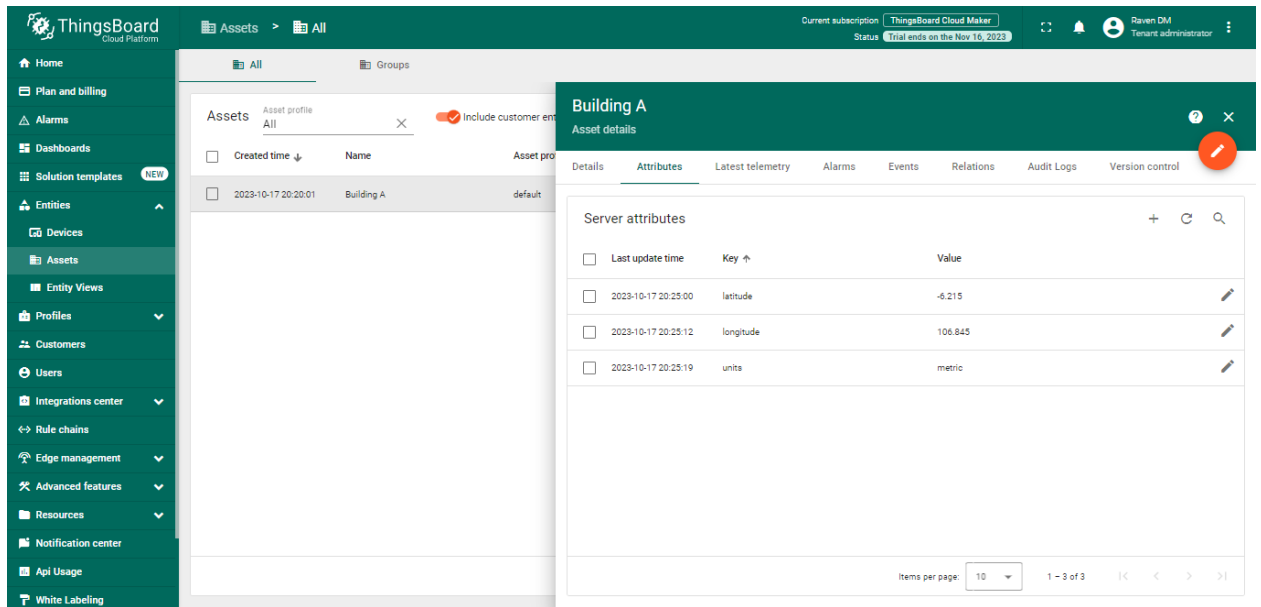


Setelah menetapkan 'Customer' klik data tersebut ke bagian attributes->add dengan pilih “+”. Tambahkan atribut dengan Field Key berisi APPID dan input API key pada value dengan tipe data string.

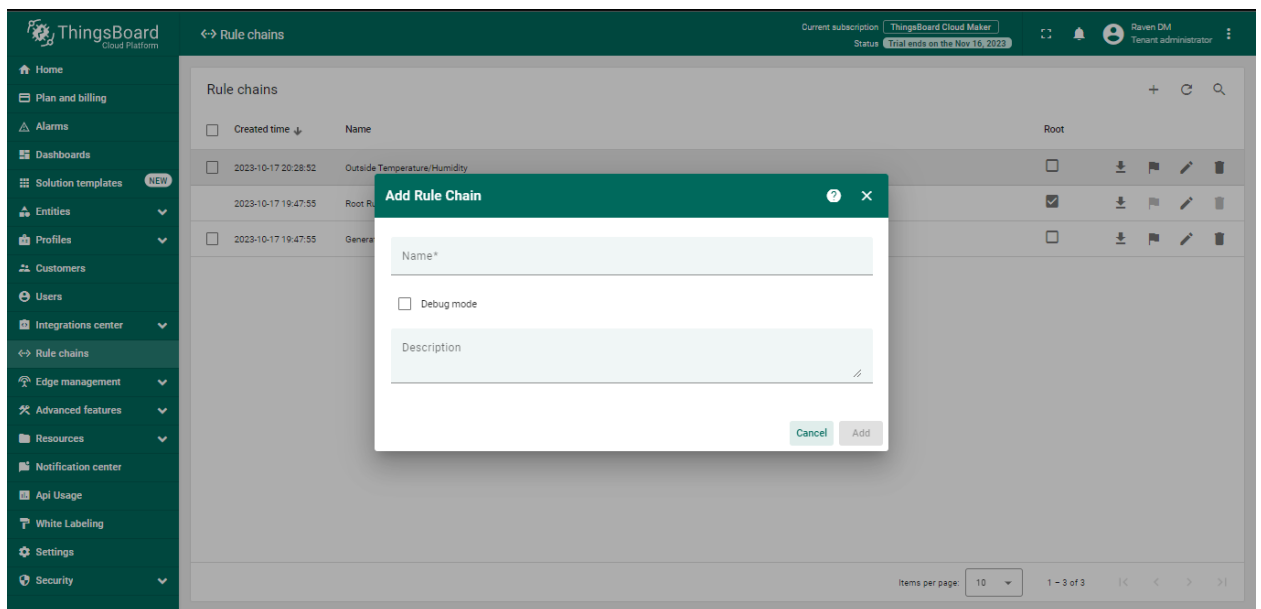


Setelah menetapkan Customers, selanjutnya beralih ke menu Entities->Assets. Kita tambahkan Nama Asset berupa Building A dengan profile default. Setelah menetapkan aset, klik profile asset tersebut selanjutnya tambahkan atribut sesuai dengan tabel dibawah ini. Pada case ini saya menggunakan koordinat yang menyatakan posisi aset Building A berada di daerah kota Jakarta, yaitu [-6.215 , 106.845].

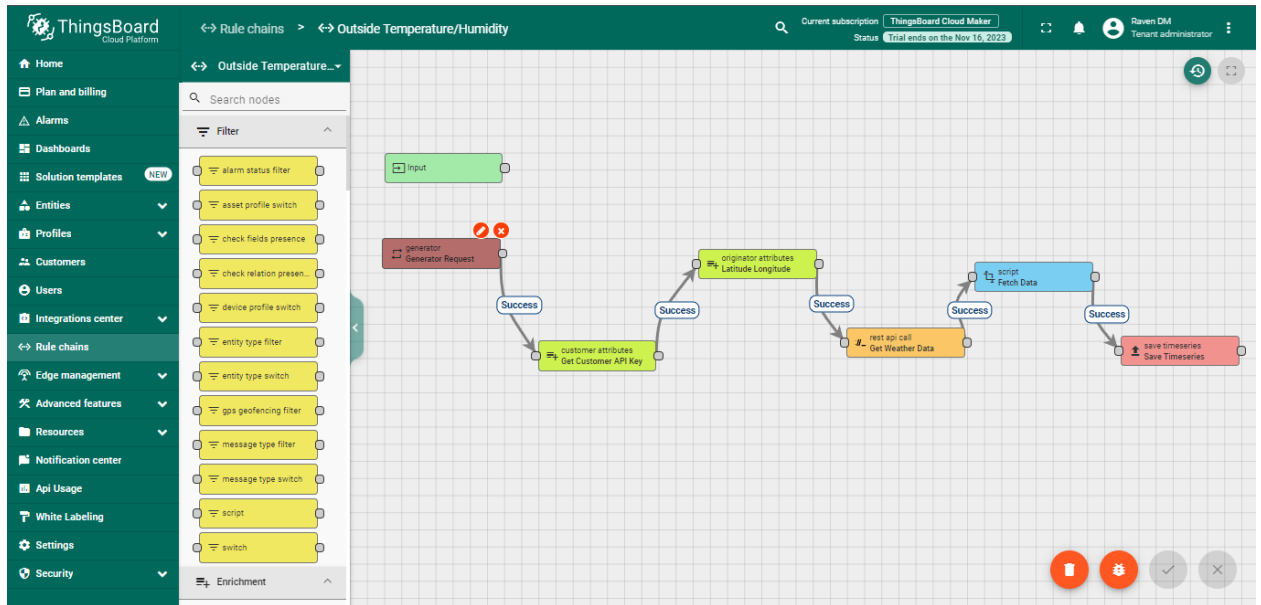
Field	Data Type	Input Data
latitude	Double	latitude of an asset
longitude	Double	longitude of an asset
units	String	"metric" for meters per second wind speed and Celsius temperature, "imperial" for miles per hour wind speed and Fahrenheit temperature, empty for meters per second wind speed and Kelvin temperature



Setelah menetapkan Customer, Aset beserta atributnya masing-masing, dapat kita lanjutkan ke bagian pemrograman untuk menampilkan data di dashboard. Untuk membuat program pilih menu Rule Chains -> add dengan pilih “+” -> Beri nama Outside Temperature/Humiditiy -> add.



Selanjutnya klik Rule Chain yang sudah ditetapkan tersebut dan laman akan beralih kepada program aliran pesan. Buat alur program seperti gambar dibawah. Untuk tipe relasi setiap node dibuat “Success”.



Penjelasan mengenai masing-masing node:

1. Node Generator

Isi nama node ini sebagai Generator Request, atur period data (panggilan API tiap x detik) sesuai kebutuhan. Selanjutnya isi bagian Originator sesuai dengan aset Building A yang sudah ditetapkan.

Kemudian pada bagian function generate diisi program yang menghasilkan pesan kosong untuk memicu panggilan REST API dengan code seperti pada gambar berikut.

Current subscription: ThingsBoard Cloud Maker
Status: Trial ends on the Nov 16, 2023
Raven DM
Tenant administrator

Generator Request

Action - generator

Details Events Help

Name*
Generator Request

Message count (0 - unlimited)*
0

Period in seconds*
60

Entity group

Originator
Type: Asset
Asset: Building A

Queue

TBEL Java Script

```
function Generate(prevMsg, prevMetadata, prevMsgType) {  
  1 return { msg: {}, metadata: {}, msgType: "POST_TELEMETRY_REQUEST" };  
}
```

2. Node pengayaan atribut customer

Node ini menempatkan atribut pelanggan APPID ke dalam metadata pesan. Isi nama node "Get Customer API Key" -> Source Attribute dan Target key diisi dengan "APPID". Keterangan detail dapat dilihat pada gambar dibawah.

Get Customer API Key

Enrichment - customer attributes

?

×

✓

×

DetailsEventsHelp

Name*Get Customer API Key

Debug mode

Mapping of customer's fields

AttributesLatest telemetry

Attributes mapping*

Source attribute keyTarget key

APPIDAPPID

Add mapping

All input fields support templization. Use `$(messageKey)` to extract value from the message and `$(metadataKey)` to extract value from the metadata.

See examples. [↗](#)

Add mapped attributes to

MessageMetadata

Rule node description

3. Node pengayaan atribut originator

Node ini akan mengambil atribut server yang sudah ditetapkan pada atribut aset Building A. Atribut yang akan dipakai adalah lintang, bujur, dan unit pencetus sebagai metadata. Isi program node

dapat dilihat pada gambar. Setelah mengisi sesuai gambar, aktifkan juga Debug mode yang berada di samping nama node.

Latitude Longitude
Enrichment - originator attributes

Details Events Help

Name*
Latitude Longitude Debug mode

Originator attributes*

All input fields support templization. Use `$(messageKey)` to extract value from the message and `$(metadataKey)` to extract value from the metadata. [See examples.](#)

Client attributes ?

Shared attributes ?

Server attributes ?
latitude × longitude × units ×

Latest telemetry ?

Add originator attributes to Message ✓ Metadata

✓ Tell failure if any of the attributes are missing

4. Node panggilan REST API

Node ini akan melakukan panggilan REST API ke openweathermap.org. Isi node ini dengan detail seperti pada gambar. Jangan lupa untuk mengaktifkan Debug mode pada node ini.

Adapun URL REST API openweathermap sebagai berikut:

[http://api.openweathermap.org/data/2.5/weather?lat=\\${ss_latitude}&lon=\\${ss_longitude}&units=\\${ss_units}&APPID=\\${APPID}](http://api.openweathermap.org/data/2.5/weather?lat=${ss_latitude}&lon=${ss_longitude}&units=${ss_units}&APPID=${APPID})

ss_latitude, ss_longitude, ss_units, ss_APPID adalah atribut server yang diambil dari metadata yang telah ditetapkan pada atribut Customers dan Asset Building A pada langkah awal.

Get Weather Data

External - rest api call

?

×

Details

Events

Help

Name*

Get Weather Data

✓

Debug mode

Endpoint URL pattern*

http://api.openweathermap.org/data/2.5/weather?lat=\${ss_latitude}&lon=\${ss_longitude}&units=\${ss_units}&APPID=\${APPID}

Use \${metadataKey} for value from metadata, \${messageKey} for value from message body.

Request method*

GET

▼

☐ Enable proxy

☐ Use simple client HTTP factory

☐ Message without quotes

If selected, request body message payload will be sent without double quotes, i.e. msg = message body

☐ Without request body

Read timeout in millis

0

The value of 0 means an infinite timeout

Max number of parallel requests

0

The value of 0 specifies no limit in parallel processing

Headers

Use \${metadataKey} for value from metadata, \${messageKey} for value from message body in header/value fields

Header	Value
--------	-------

5. Node transformasi skrip

Node ini akan menempatkan rata-rata suhu luar, suhu maksimal, suhu minimal, dan rata-rata kelembaban ke dalam pesan.

Isi code sebagai berikut serta aktifkan Debug Mode:

```
var newMsg = {
  "outsideTemp": msg.main.temp,
  "outsideMaxTemp": msg.main.temp_max,
  "outsideMinTemp": msg.main.temp_min,
  "outsideHumidity": msg.main.humidity,
};

return {msg: newMsg, metadata: metadata, msgType: msgType};
```

Fetch Data

Transformation - script

Details

Events

Help

Name*

Fetch Data

Debug mode

TBEL

Java Script

function Transform(msg, metadata, msgType) {

1 var newMsg = {

2 "outsideTemp": msg.main.temp,

3 "outsideMaxTemp": msg.main.temp_max,

4 "outsideMinTemp": msg.main.temp_min,

5 "outsideHumidity": msg.main.humidity,

6 };

7

8

9 return {msg: newMsg, metadata: metadata, msgType: msgType};

}

Test transformer function

Rule node description

6. Node penyimpan rentang waktu

Node ini akan memasukan pesan ke telemetry.

dasht
dasht

Save Timeseries

Action - save timeseries

Details

Events

Help

Name*

Save Timeseries

Debug mode

Default TTL in seconds*

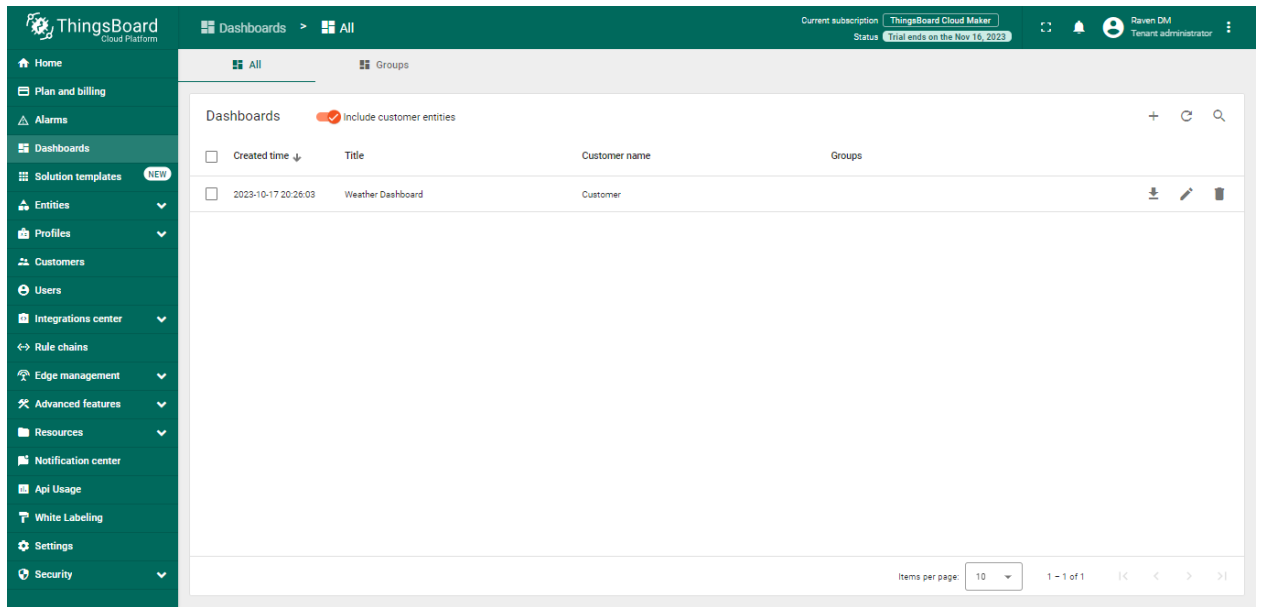
0

☐ Skip latest persistence

☐ Use server ts

Enable this setting to use the timestamp of the message processing instead of the timestamp from the message. Useful for all sorts of sequential processing if you merge messages from multiple sources (devices, assets, etc).

Rule node description



Kemudian akan masuk ke laman dashboard yang masih kosong. Pada langkah ini dapat menambahkan widget sesuai keinginan karena banyak jenis widget yang disediakan oleh platform ThingsBoard. Pada paper ini kita gunakan widget open map, label aset, grafik suhu, dan grafik kelembababn. Data yang ditampilkan juga dapat dianalisis secara real-time dan history-time.

