

# Utilisation de la plateforme [maker.allthingstalk.com](http://maker.allthingstalk.com)

# maker.allthingstalk.com

- Plateforme destinée à l'IOT
  - Comprend un api REST et un broker MQTT
  - L'api permet la communication entre différentes applications et la plateforme
    - exemples : authentication, récupération de données, création et suppression de capteurs...
- Le broker mqtt permet aux objets de communiquer avec la plateforme

# maker.allthingstalk.com

- Se créer un compte
- Télécharger et installer les bibliothèques (<http://docs.allthingstalk.com/developers/sdk/arduino/> )

# Installer les librairies depuis le fichier

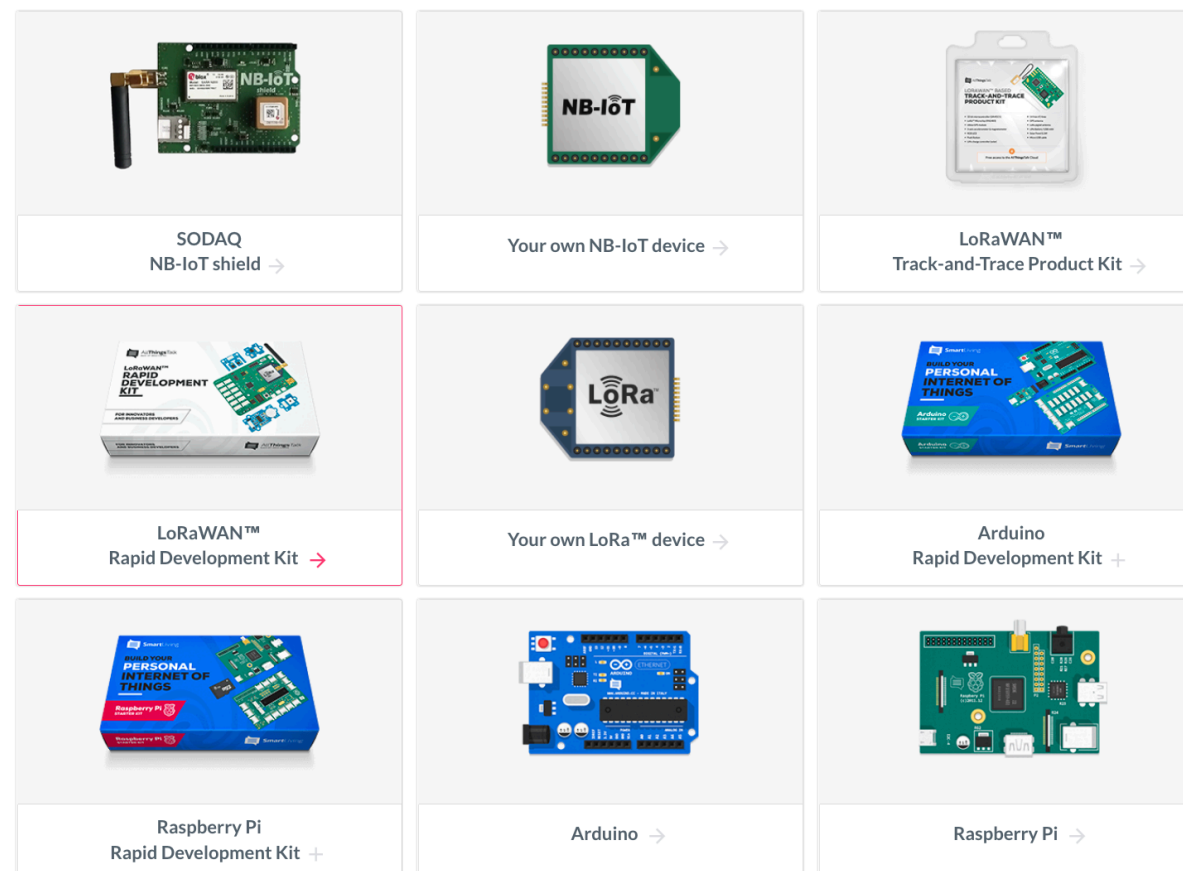
- Décompresser le fichier **arduino-sdk-master.zip**
- copier les 3 sous-répertoires contenus dans **libraries** (pubsubclient, ArduinoJson et ATT\_IOT) vers le sous-répertoire libraries du répertoire Arduino de votre utilisateur

# Créer un device

Dans le “ground” par défaut (Playground) cliquer sur

+ CONNECT A DEVICE

Choisir Arduino dans la liste



# Créer un device

- Le device représente un microcontrôleur (ou un téléphone, ou une application web)
- Le device peut avoir plusieurs “assets” qui représentent des capteurs ou des actuateurs
- Il est possible de créer des assets programmatiquement ou via l’interface

# Créer un asset

- Un asset représente un capteur ou un actuateur
- il est défini principalement par 3 éléments :
  - un identifiant (*String* : sera utilisé pour identifier l'asset lorsque l'on veut communiquer avec lui)
  - un titre (*String* : sera affiché dans l'interface)
  - un type : String, integer, number, boolean
- Le type de l'asset définit les valeurs qu'il peut prendre

# Publier des données

- Exemple : voir code (nb: le code se base sur l'utilisation de la librairie WiFi101)
- récupérer le deviceId et le deviceToken dans l'interface et les copier dans le fichier arduino\_secrets.h



# Publier des données

 my-iot-device |  You

[+ CREATE PINBOARD](#) [⚙ SETTINGS](#)

Click the **CREATE PINBOARD** button in the upper right corner to visualise the device data.


### SETTINGS

**my-iot-device**  
my-iot-device


Generic Arduino device

>


MANAGEMENT

 **Authentication**  
View your Device Tokens

>

 **Connectivity**  
Activate device on supported LPWAN

>

 **Delete device**  
This action cannot be undone.

>

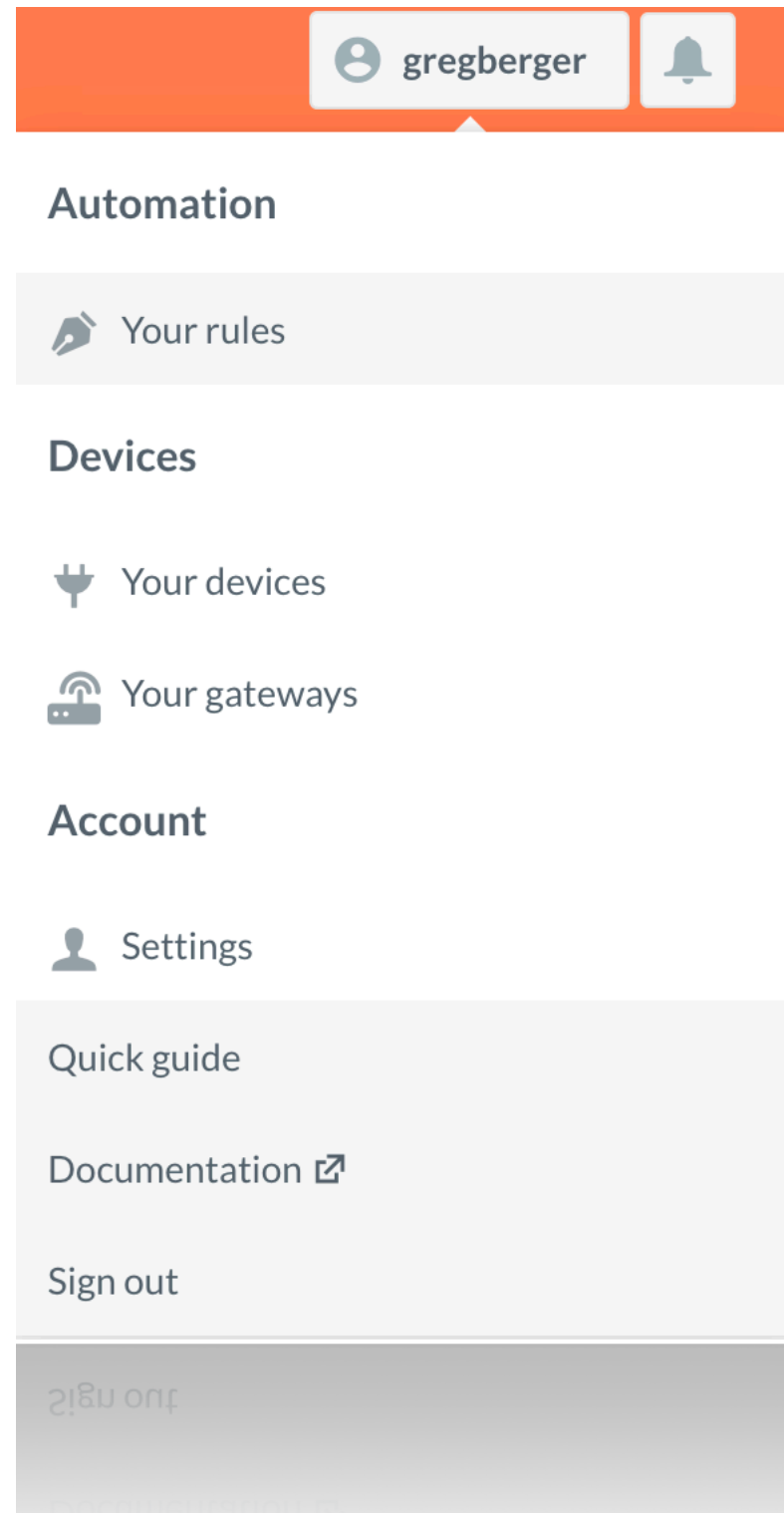
# Recevoir des données

- Voir code exemple

# Créer des règles

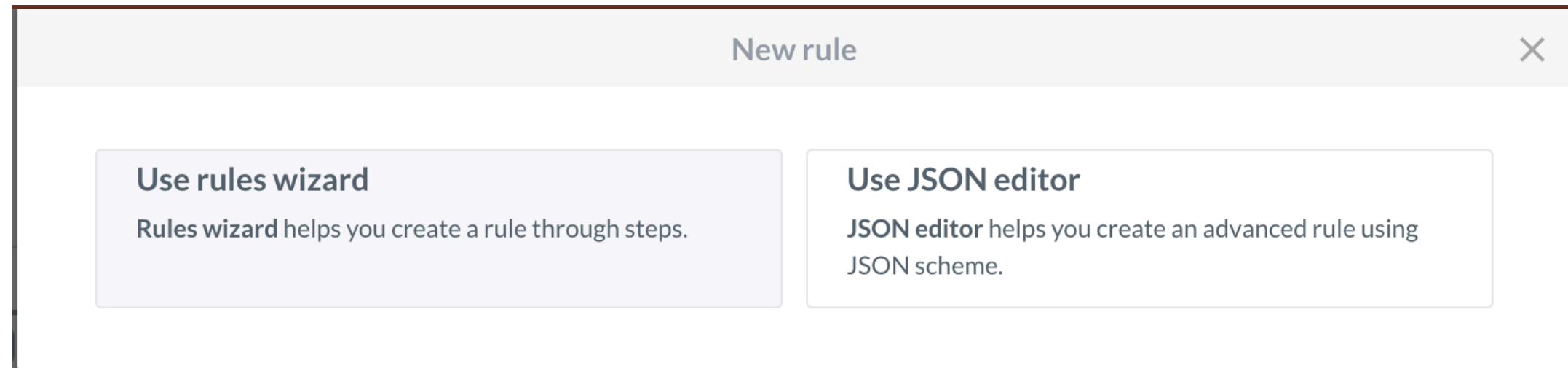
- La plateforme permet de créer des règles pour :
  - recevoir des notifications en fonction de la valeur d'un capteur
  - faire interagir des devices entre eux (exemple Smartphone et LED)

# Créer des règles



# Créer des règles

## Utiliser le wizard



# Créer des règles

Sélectionner le déclencheur : un asset ou un déclencheur périodique

When

Then

Select a trigger



Day and time of the week

Select day and time to trigger the rule



Asset state change

Select an existing asset to trigger the rule

Select a trigger

# Créer des règles

Choisir le device, l'asset et le déclencheur

When

Your smartphone

Shake

Then

..

Choose a device

Your smartphone

Choose an asset

Shake

Set a condition

On every state change

Compare to some value

Compare to other asset

# Créer des règles

Sélectionner l'action à effectuer lorsque la condition est remplie

When

Your smartphone

Movement==moving left



Then



Select Action



Notify me

You will be notified by the selected notification channels (web, push or e-mail)



Actuate an asset

Select an existing asset as rule action



# Créer des règles

Si il s'agit d'un asset, définir la valeur à envoyer

When

Your smartphone

Movement==moving left

●●

✓

Then

my-iot-device

led=true

●●

✓

✕

Choose a device

my-iot-device

▼

Choose an asset

led

▼

Set to

Set to some value

✓

Set to other asset value

Enter value to set

True

✓

False

Expects type: boolean

# Créer des règles

- Rappel : le type de l'asset détermine le type de valeur que l'on pourra recevoir ou envoyer
- exemple: si un asset est défini comme boolean, on ne pourra recevoir ou envoyer que **true** ou **false**