IoT Platforms

Frank Walsh

What are IoT platforms

- IoT applications combine sensors, devices, data, analytics and integrations in a seamless and unified way
  - e.g. your project!
- IoT Platforms provide software tools and components to:
  - connect sensors, devices, and data networks
  - Analyse and store data
  - Integrate with other apps
- So what? We know the tech for that now (I2C, SPI, BLE, MQTT, Python...)
- Main selling point of an IoT platform is software that it
  - accelerates the IoT development process
  - Focuses on IoT: brings in best of breed features
  - Provides initial scaffolding for IoT projects

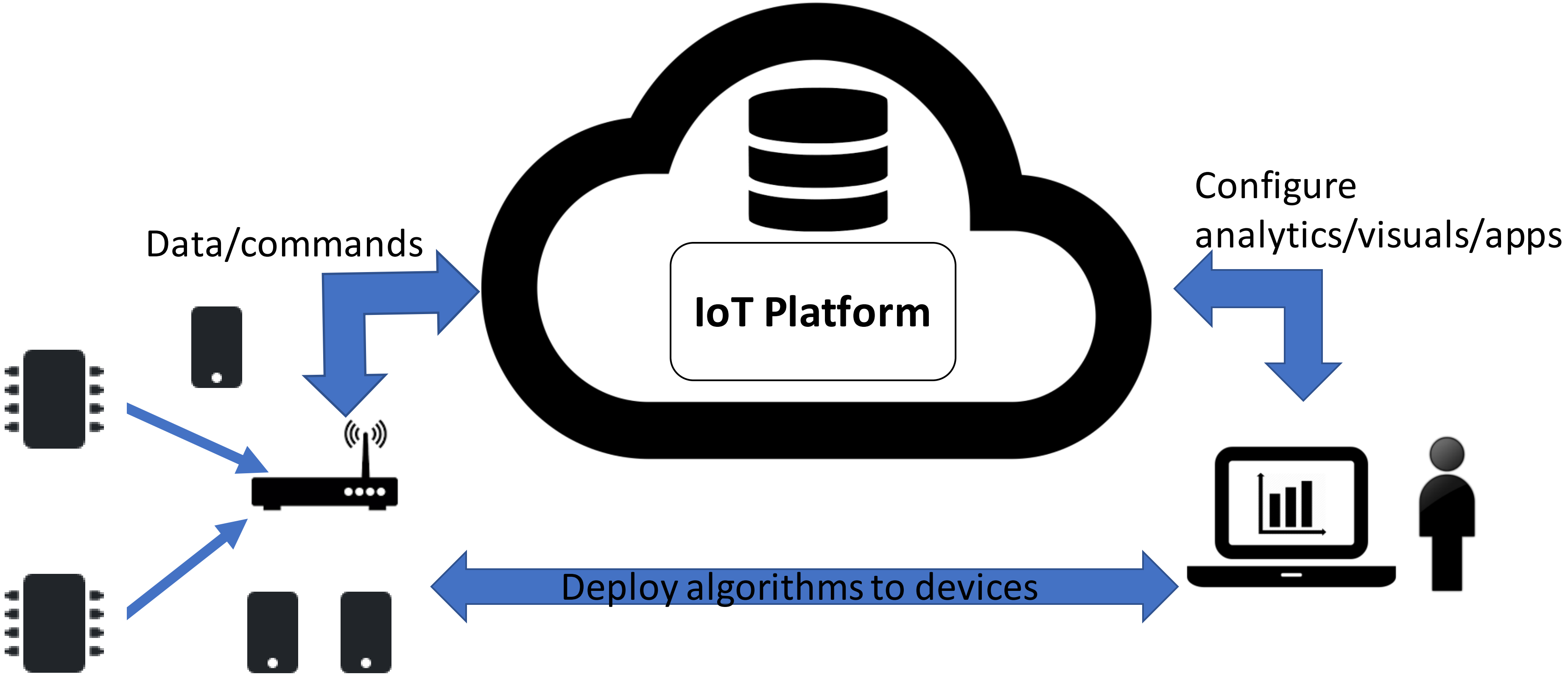
What are IoT Platform

- Many(not all) are cloud-based platforms that require subscription
- Provide device/language agnostic set of Software Development kits
  - Arduino/RPi/beagleboard
- IoT development is generally iterative:
  - Starts with initial simple use case
  - Once operational, data/insights result in new usecased
- IoT platforms should promote scalable, iterative development
  - Allow for quick app development
  - Ability to adapt/optimise apps quickly

IoT Platform Characteristics

- Manage many concurrent device connections
- Connectivity across several connection types
- "Off-the-peg" IoT protocol stack
- Manage/analyse/visualise data
- Integrations to other services/apps
- App Development

IoT Platform – generalised



Data/commands

IoT Platform

Configure analytics/visuals/apps

Deploy algorithms to devices

IoT Platform Advantages

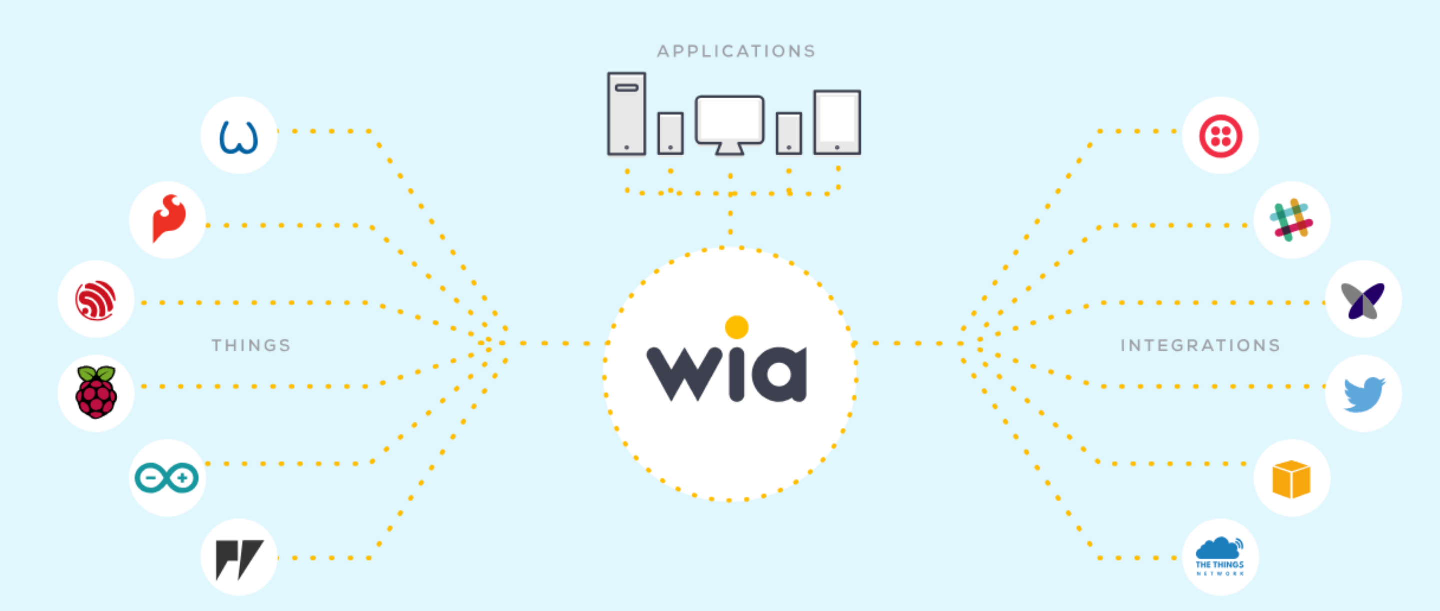
- Sofware components that has been pre-built and pre-tested. This increases the reliability of your application and reduces development effort.
- IoT frameworks constantly evolve, providing new features, integrations etc.
- Encourages better "design pattern" for your IoT app.
- Predefined APIs and docs
  - Great for collaboration
- "Baked-in" standards and features:
  - Security, authentication, scalability...

Which one?

- Connectivity
  - Does the platform provide suitable capability and integrations (WiFi/Cellular/LPWan-Sigfox)
- Maturity
  - In business for long? Critical mass in developer community?
- Free
  - Is there a free tier (handy for evaluation)?
- Service type
  - Platforms try to distinguish themselves – what specialisms/USP does it have?
- Security
  - What security model do they use? Is there security issues reported in past?
- Geographic area
  - Does it operate well at your location (can you select edges/data centres)

Wia.io

- "Enabling devices to communicate with one another in a simple, easy way."
- "We take care of the messy cloud infrastructure and expose a globally available cloud API that developers can interact with to build intelligent and complex applications."



"Any device. Any application. One cloud."

Wia Overview

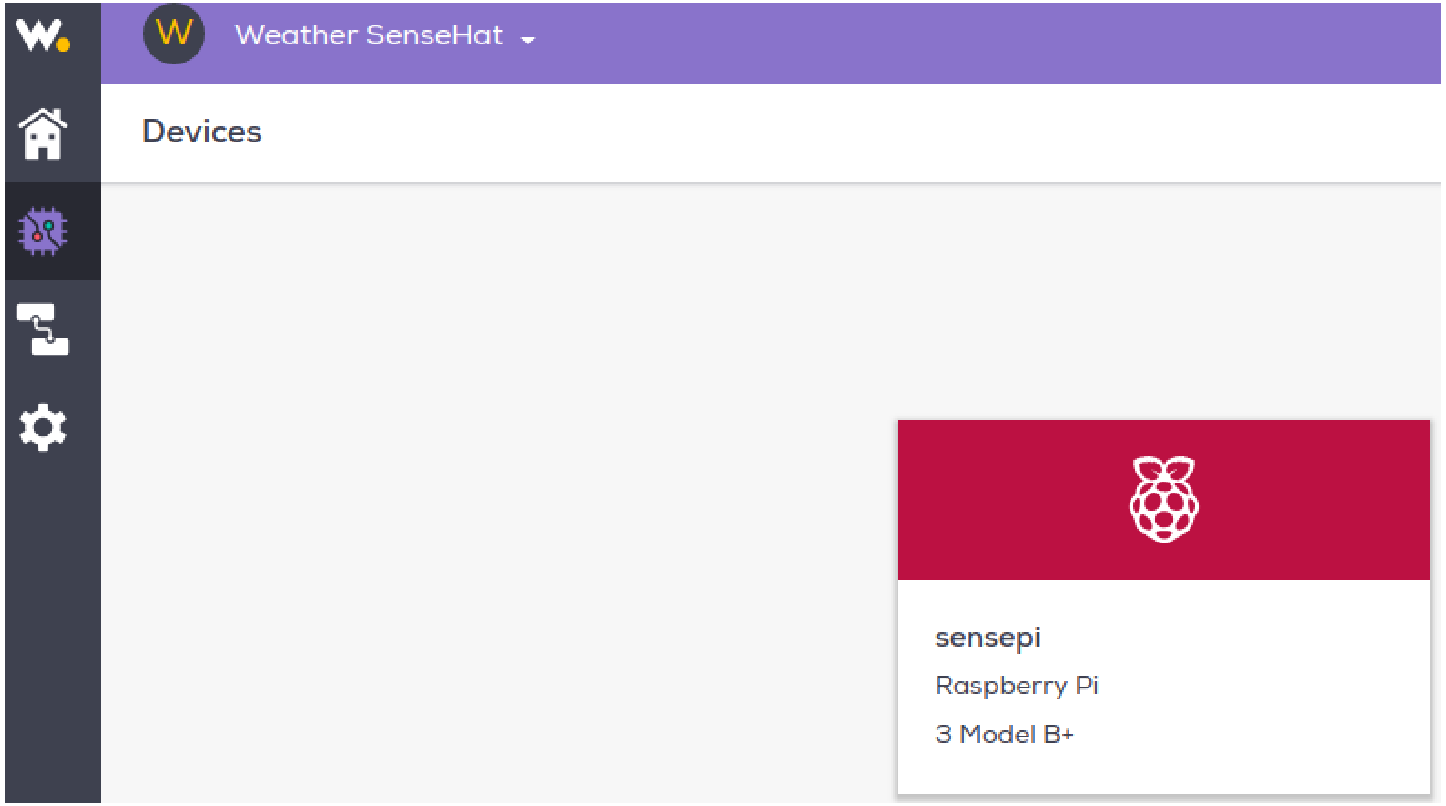
- Account-based
    - Must create an account to use.
- Provides mechaism to connect devices
    - Easiest to use relevant SDK for device/language.
- Specialism: 3rd party integrations. Nice programming abstrations (means you can connect with minimal code/effort)
- Provides REST/MQTT APIs (although REST API still under development)

Wia architecture/Terminology

- Create a "Space":
  - Contains your devices
- Add Device(s)
  - Add a device(e.g. Raspberry Pi)
- Start publishing "Events" from device
  - e.g. door opened, temperature
- View/analyse data with widgets
  - Dashboard for your data
- Create a "Flow"
  - Connect 3rd party services and/or implement logic functions.
- Command your device
  - Control your device (e.g. rotate motor, turn on device) using Commands

Wia Example - Create Device

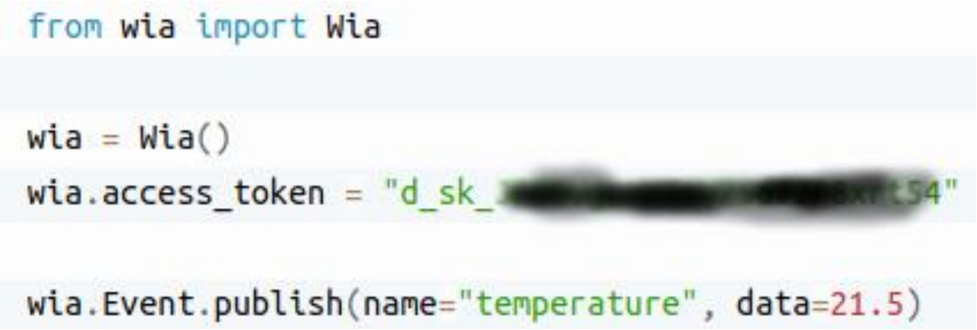
- Configure a device in a space
- Identified by unique Device ID (e.g. Dev_fxxYYddfsf)
- Assigned Device Secret Key (e.g. d_sk_fdsbjkb32423fnsjf)
  - Used to interact with API



sensepi
Raspberry Pi
3 Model B+

Wia Example - Publish Event

- Program the device (using Python):

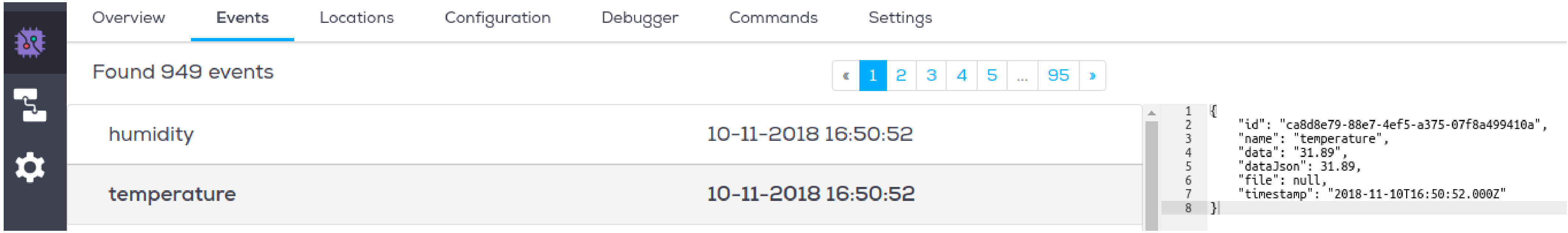


```
from wia import Wia

wia = Wia()
wia.access_token = "d_sk_[redacted]"

wia.Event.publish(name="temperature", data=21.5)
```

- See Events pulished to device on Wia



Found 949 events

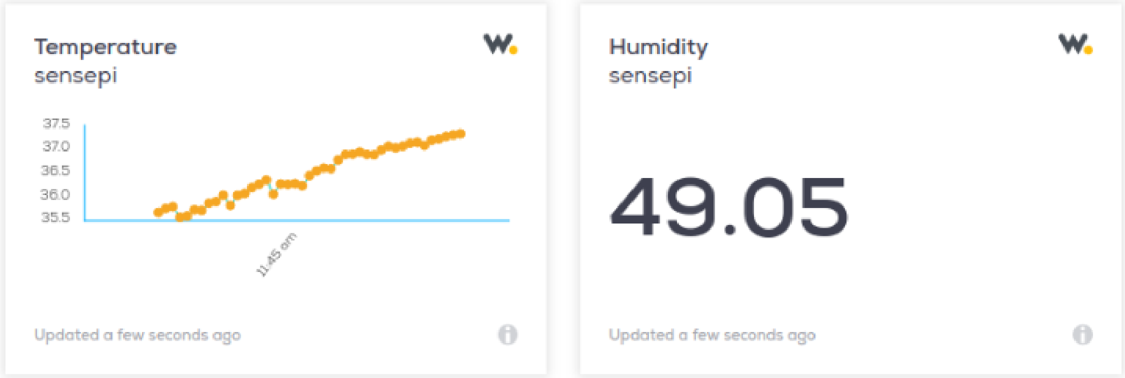
```
humidity        10-11-2018 16:50:52
temperature     10-11-2018 16:50:52
```

```
{
    "id": "ca8d8e79-88e7-4ef5-a375-07f8a499410a",
    "name": "temperature",
    "data": "31.89",
    "dataJson": 31.89,
    "file": null,
    "timestamp": "2018-11-10T16:50:52.000Z"
}
```

Wia Example - View Data

- Create a Widget for an event
- Several widgets:
  - Text, photo, location, graph
- Embeddable in other web apps:

Wia Weather Station



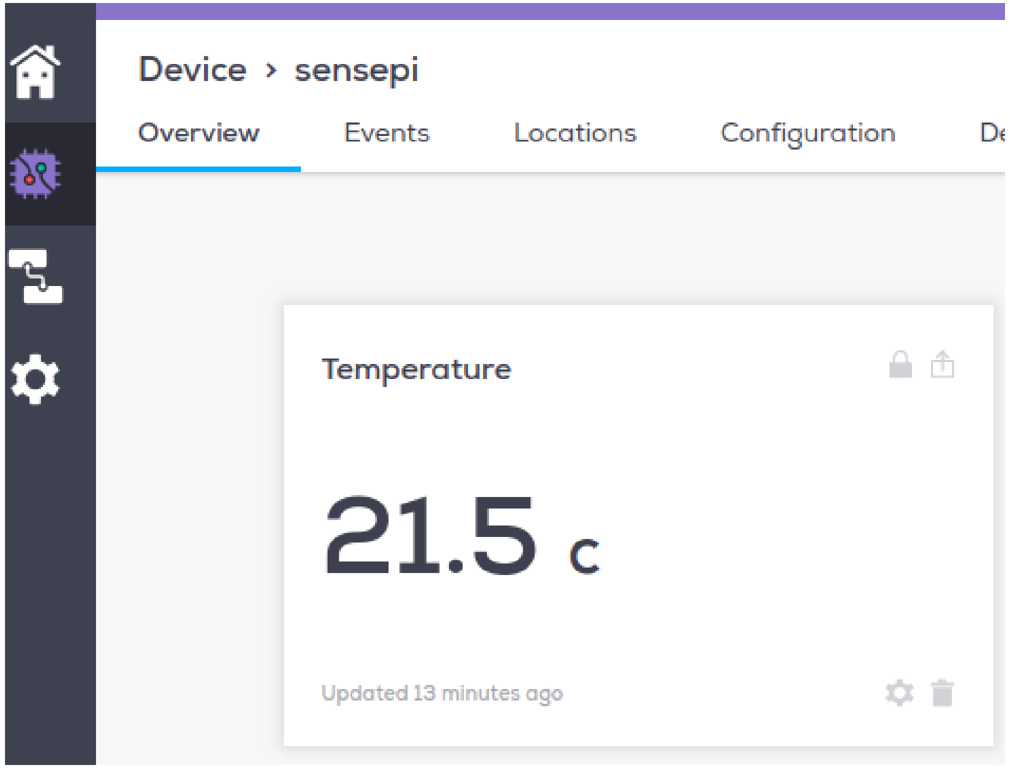
Temperature
sensepi

Updated a few seconds ago

Humidity
sensepi

49.05

Updated a few seconds ago



Device › sensepi

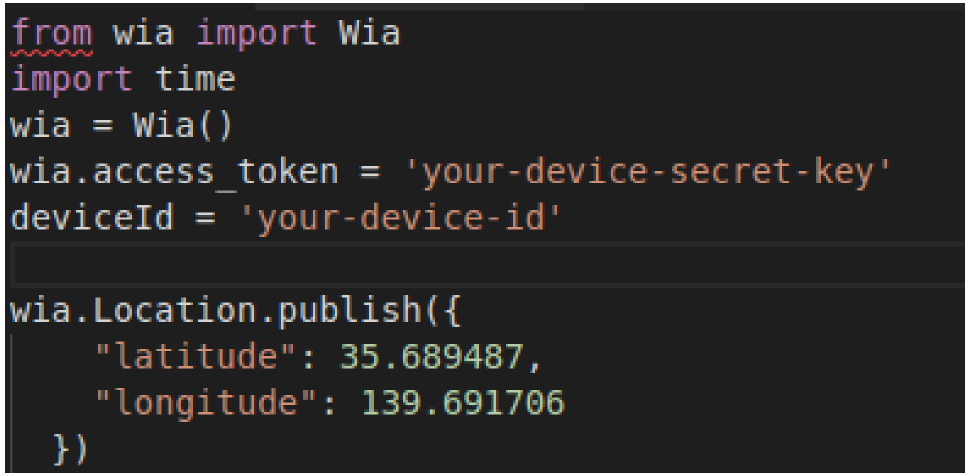
Temperature

21.5 c

Updated 13 minutes ago

Wia Example - Location

- Built-in Location API that makes it easy to track devices
- Use location function to publish/subscribe to lat/long data
- Like other widgets, can embed in other apps.

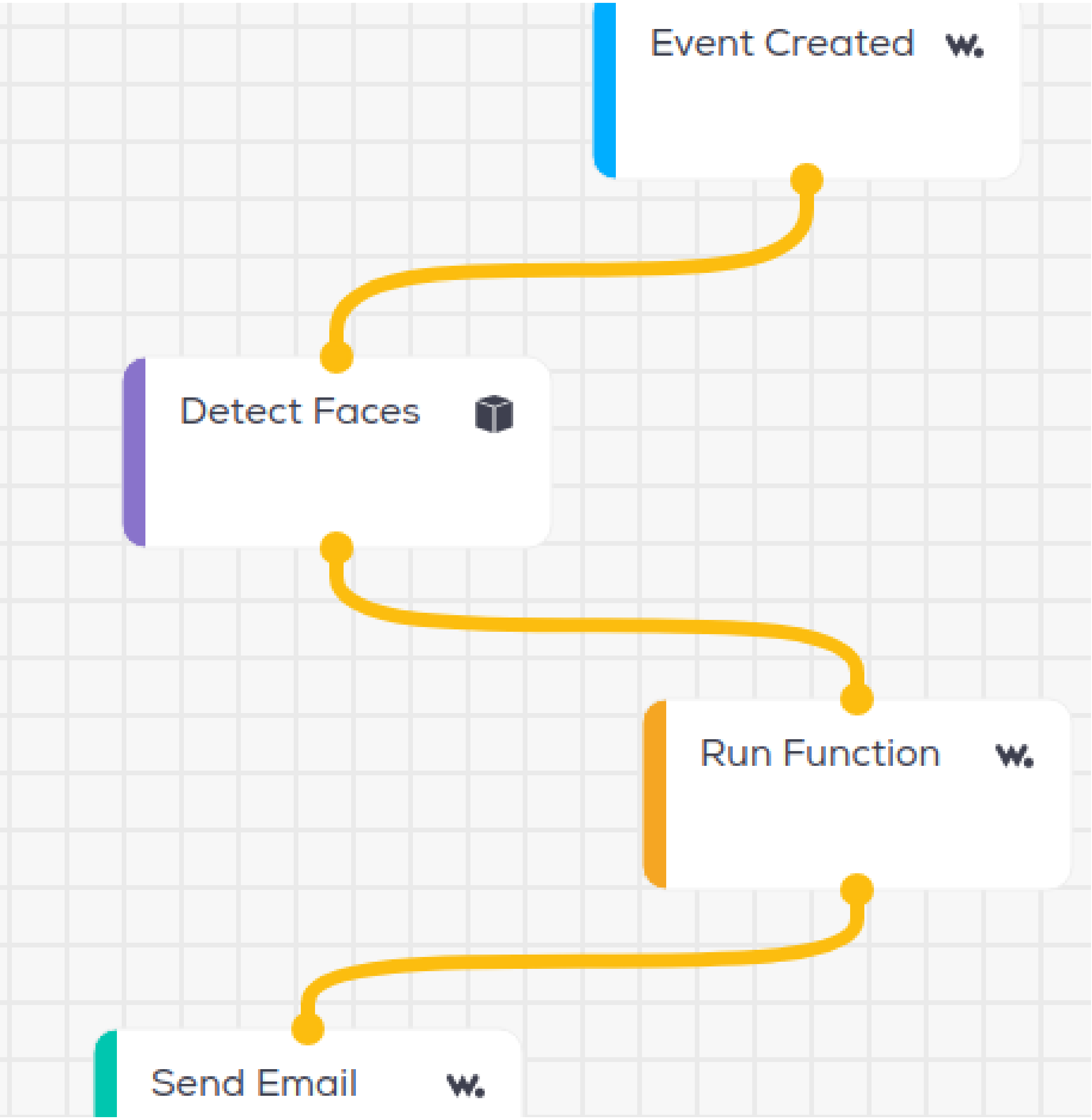


```
from wia import Wia
import time
wia = Wia()
wia.access_token = 'your-device-secret-key'
deviceId = 'your-device-id'

wia.Location.publish({
    "latitude": 35.689487,
    "longitude": 139.691706
  })
```

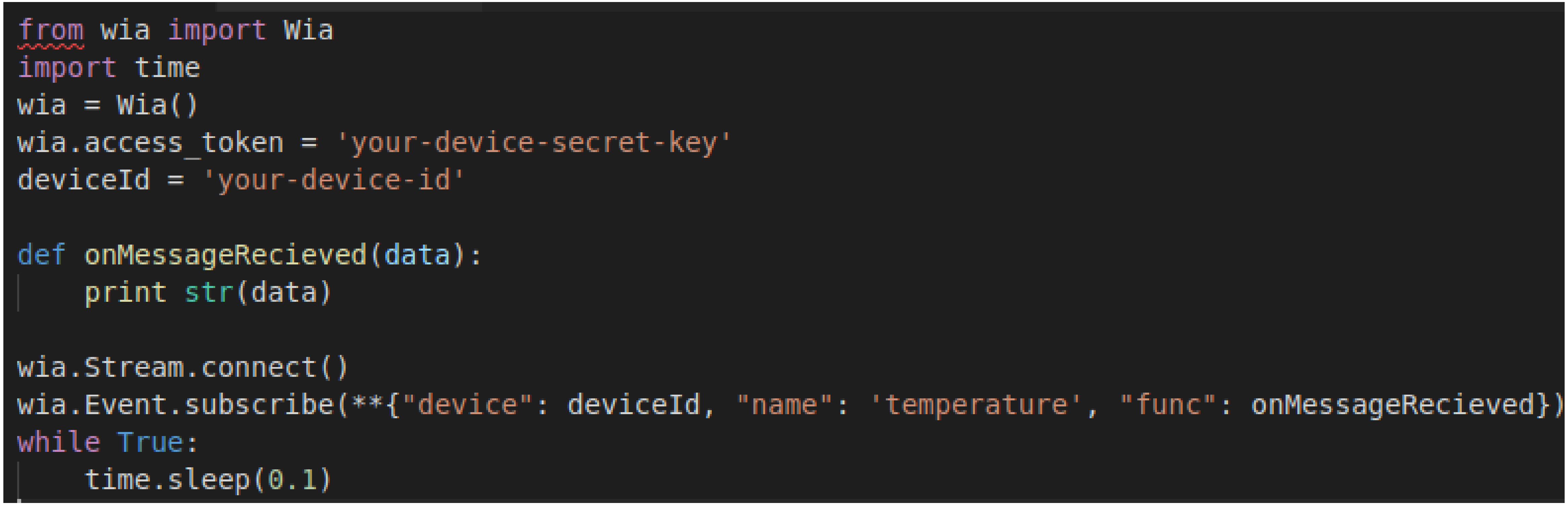
Flows

- Use Flow Builder to connect devices to other services
- Use flows to implement limited logic/analysis/decisions (only in Javascript though)
- Example, use Amazon Face Rekognition integration to detect a smile.



Rest/MQTT APIs

- Can access events/devices and data using the APIs (just like last week)
- Example: sunscribe to "temperature" event on device



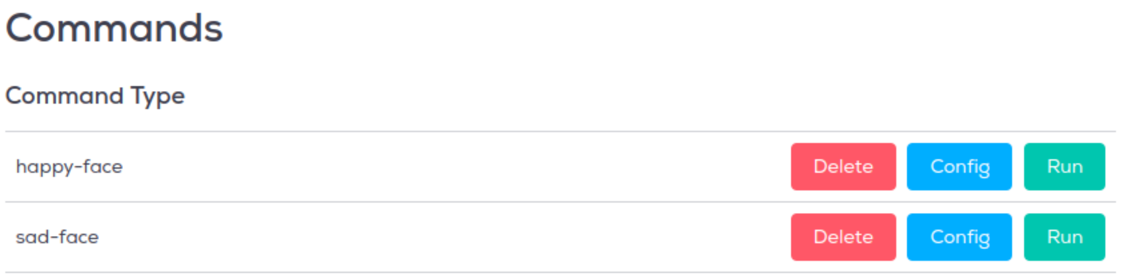
```python
from wia import Wia
import time
wia = Wia()
wia.access_token = 'your-device-secret-key'
deviceId = 'your-device-id'

def onMessageRecieved(data):
    print str(data)

wia.Stream.connect()
wia.Event.subscribe(**{"device": deviceId, "name": 'temperature', "func": onMessageRecieved})
while True:
    time.sleep(0.1)
```

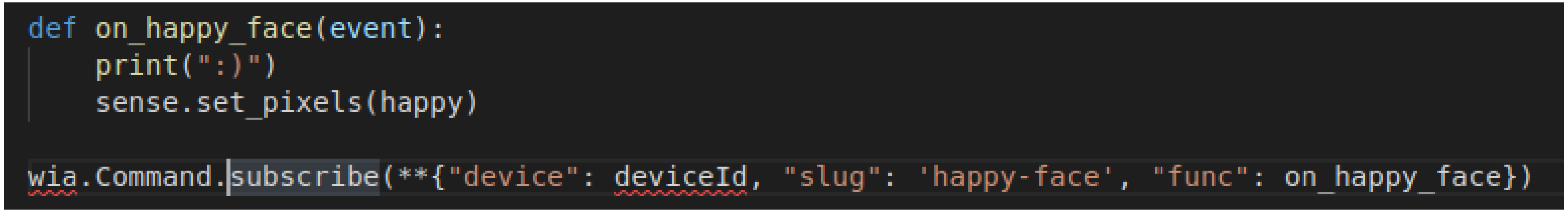
Wia Example - Commands

- Used to run code/actuate something on a device.
- Create Command for device in Wia.
  - Command is associated with a "slug"
- Devices can subscribe to a command using slug name



Commands

Command Type
happy-face
sad-face



```
def on_happy_face(event):
    print(":)")
    sense.set_pixels(happy)

wia.Command.subscribe(**{"device": deviceId, "slug": 'happy-face', "func": on_happy_face})
```

- Display happy emoticon when happy-face command is published; Control electric motor...

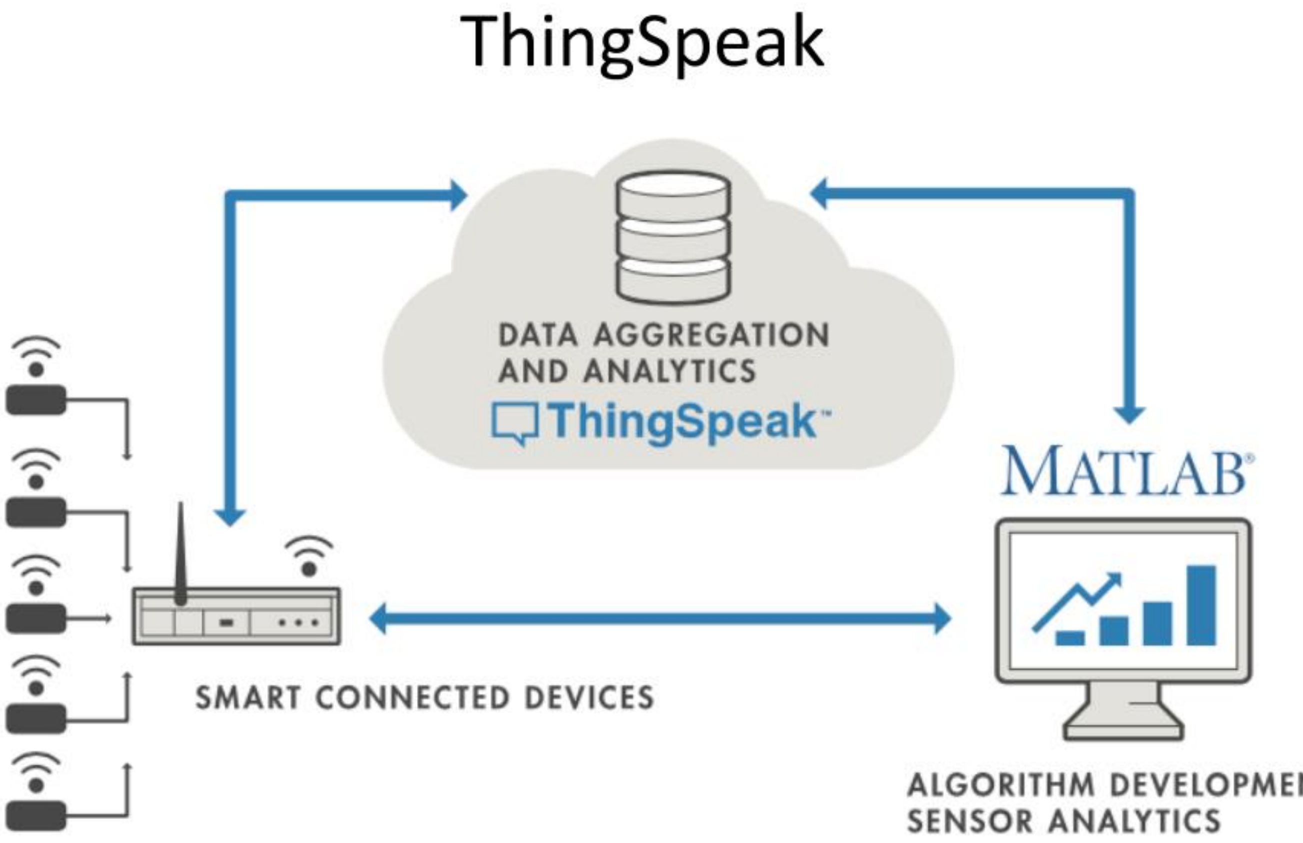
Thingspeak

Thingspeak

- **ThingSpeak** is a cloud-based IoT platform to store and retrieve data from devices.
  - Uses HTTP protocol/Restful APIs
- "Collect and analyse data quickly and easily"

Thingspeak Overview

- Account-based
    - Can create free account online
- Brought to you by the people who made Matlab
    - Uses Matlab features/toolboxs
- SDKs/librarys for popular languages/devices
- Restful API means should work with any device



Thingspeak – basic use

Create a new channel

- Channels collect data

Collect data in the channel

- Devices write data to channels

Analyse the data

- Run analytical algorithms/visualise your data

Act on the data

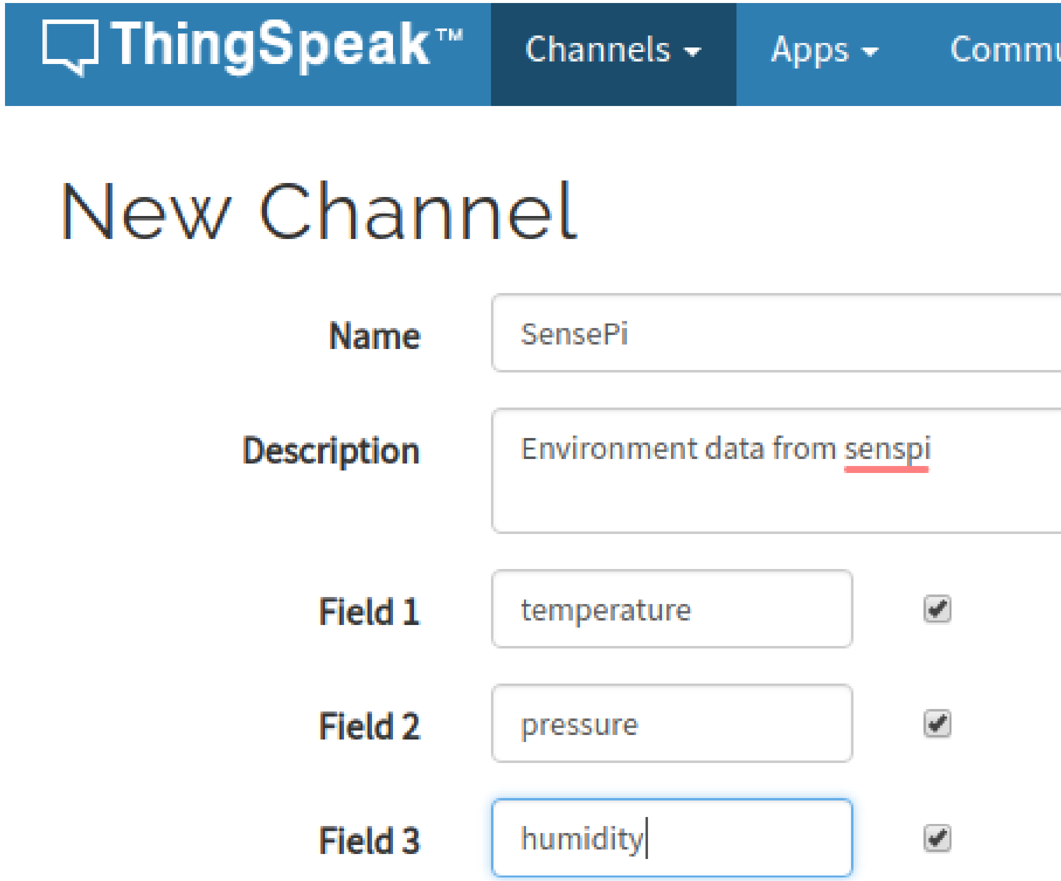
- Test for certain conditions and perform actions

ThingSpeak – Create new channel

CREATE A NEW CHANNEL TO COLLECT DATA FROM DEVICES

DEFINE DATA FIELDS FOR THE CHANNEL(MAX 8)

CAN ALSO INPUT LOCATION(LAT/LONG) OF CHANNEL SOURCE)

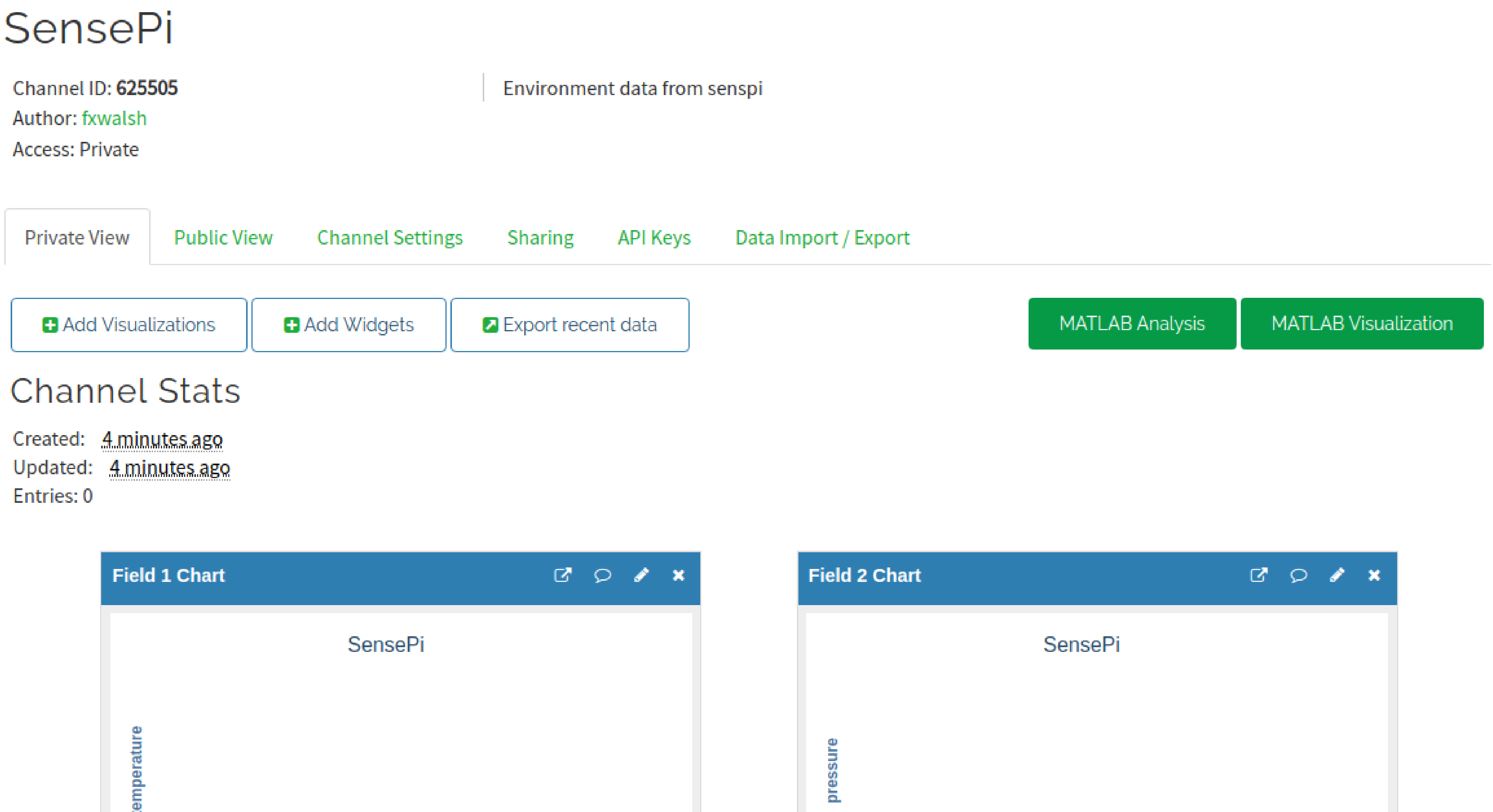


New Channel

Name	SensePi	
Description	Environment data from senspi	
Field 1	temperature	☑
Field 2	pressure	☑
Field 3	humidity	☑

Thingspeak - New channel

- Once saved you can access channel page:



SensePi

Channel ID: **625505**
Author: fxwalsh
Access: Private

Environment data from senspi

Channel Stats

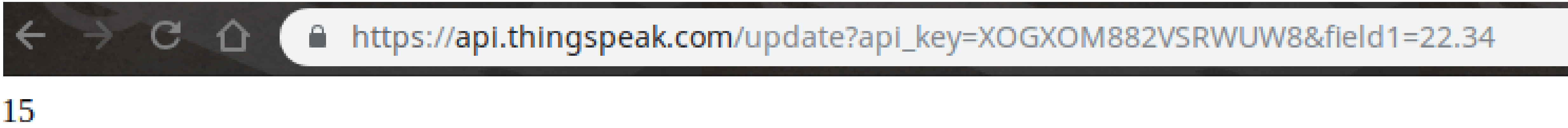
Created: 4 minutes ago
Updated: 4 minutes ago
Entries: 0

Thingspeak - Add data to channel

- Programmatically, many ways!
  - Construct HTTP GET request and include field values in query string

GET https://api.thingspeak.com/update?api_key=<WRITE-KEY>&field1=12

- Because always HTTP GET request, can test from a browser:

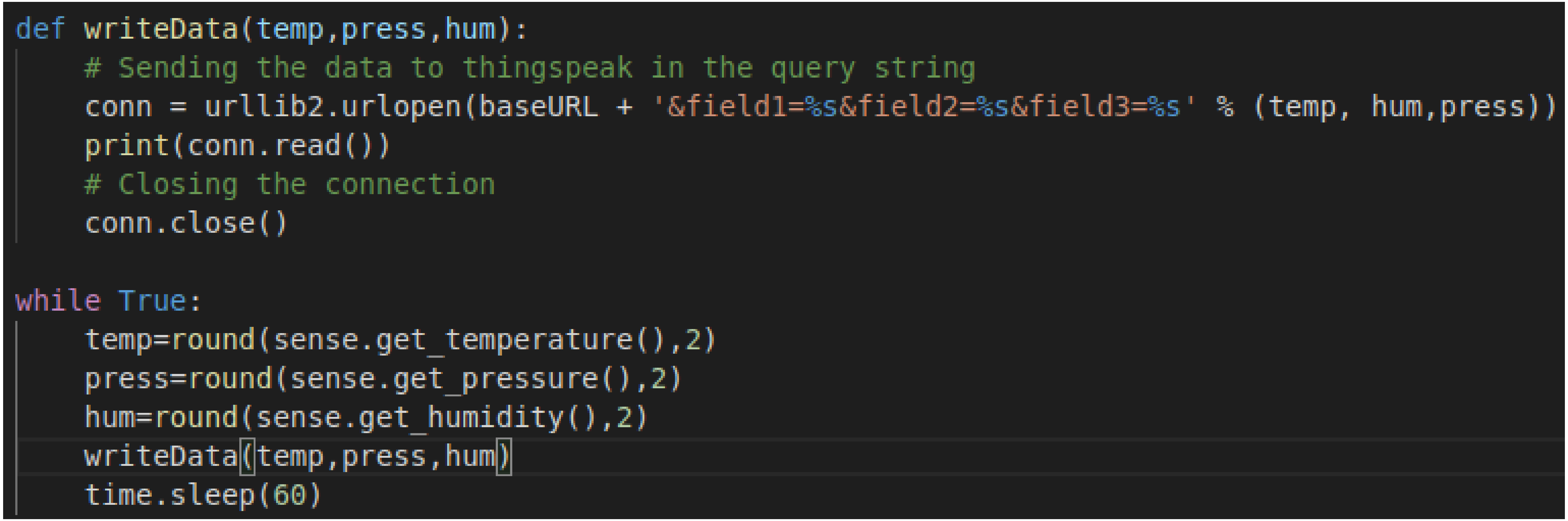


https://api.thingspeak.com/update?api_key=XOGXOM882VSRWUW8&field1=22.34

```
15
```

Thingspeak – Add data with python

- Make HTTP request from Python:

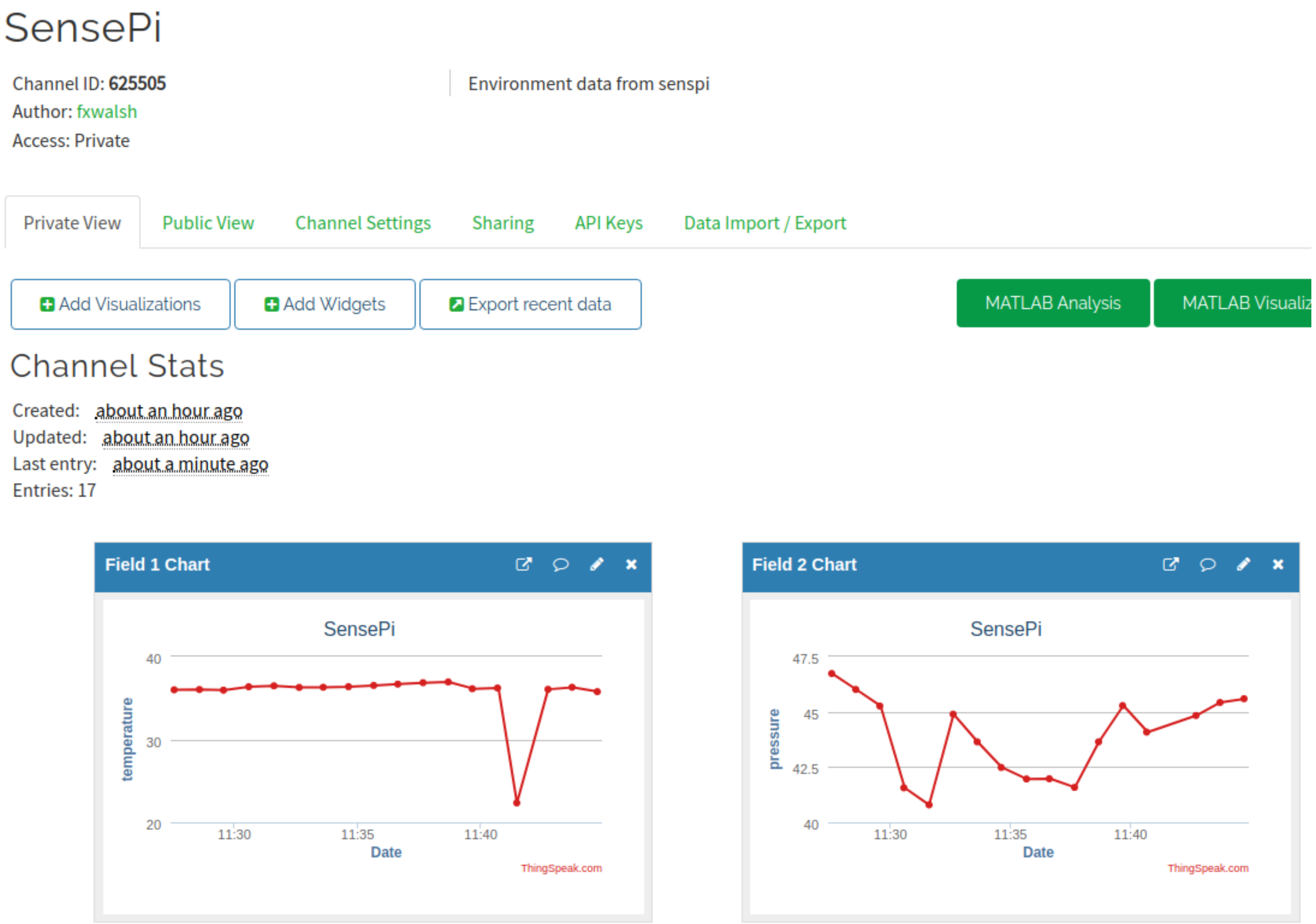


```
def writeData(temp,press,hum):
    # Sending the data to thingspeak in the query string
    conn = urllib2.urlopen(baseURL + '&field1=%s&field2=%s&field3=%s' % (temp, hum,press))
    print(conn.read())
    # Closing the connection
    conn.close()

while True:
    temp=round(sense.get_temperature(),2)
    press=round(sense.get_pressure(),2)
    hum=round(sense.get_humidity(),2)
    writeData(temp,press,hum)
    time.sleep(60)
```

Think Speak – Analyse data

- Thingspeak will visualise each field by default in channel view



SensePi

Channel ID: **625505**
Author: fxwalsh
Access: Private

Environment data from senspi

Private View | Public View | Channel Settings | Sharing | API Keys | Data Import / Export

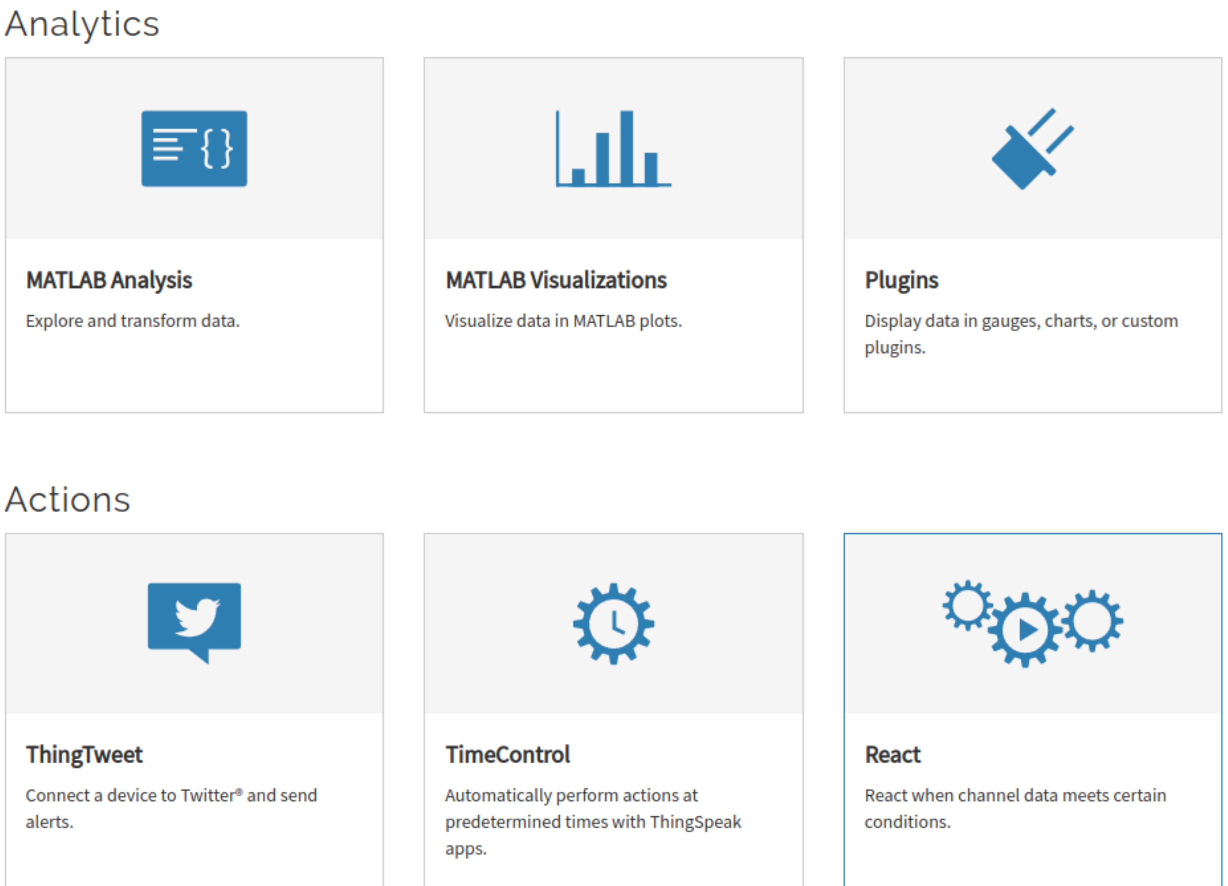
Add Visualizations | Add Widgets | Export recent data | MATLAB Analysis | MATLAB Visualiz

Channel Stats

Created: about an hour ago
Updated: about an hour ago
Last entry: about a minute ago
Entries: 17

Thingspeak - Apps

- The Apps tab provides various mechanism to transform, analyse,visualise and act on data.



Analytics

MATLAB Analysis

Explore and transform data.

MATLAB Visualizations

Visualize data in MATLAB plots.

Plugins

Display data in gauges, charts, or custom plugins.

Actions

ThingTweet

Connect a device to Twitter® and send alerts.

TimeControl

Automatically perform actions at predetermined times with ThingSpeak apps.

React

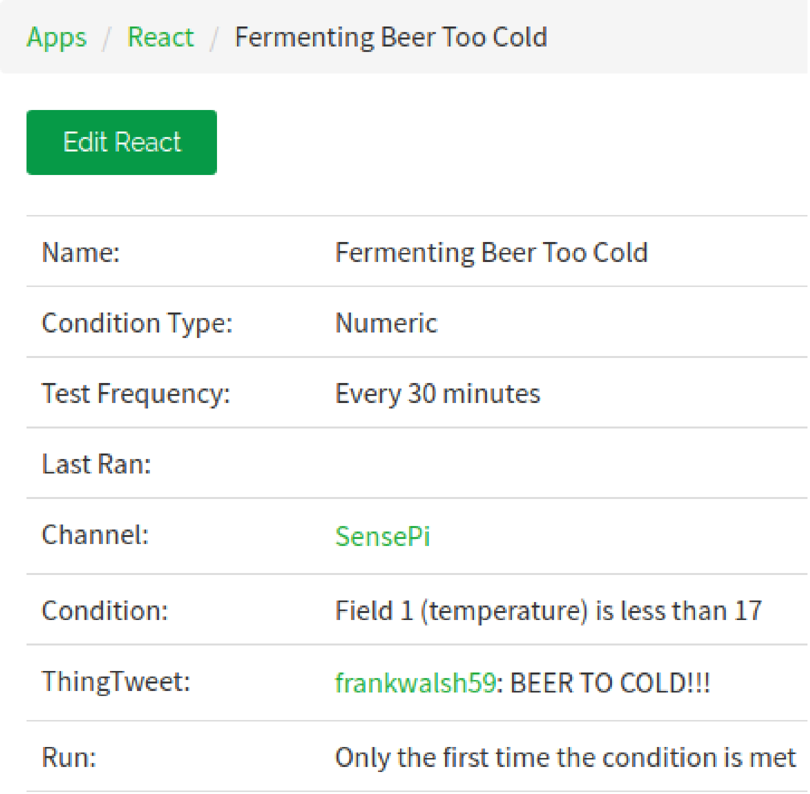
React when channel data meets certain conditions.

ThingSpeak Example: ThingTweet

- Link Twitter account to Thingspeak
- Create a **React** to tweet when a certain condition is met.
- Also tweet from device using HTTP POST:

POST https://api.thingspeak.com/apps/thingtweet/1/statuses/update

api_key=<YOUR_API_KEY>
status=I just posted this from my thing!



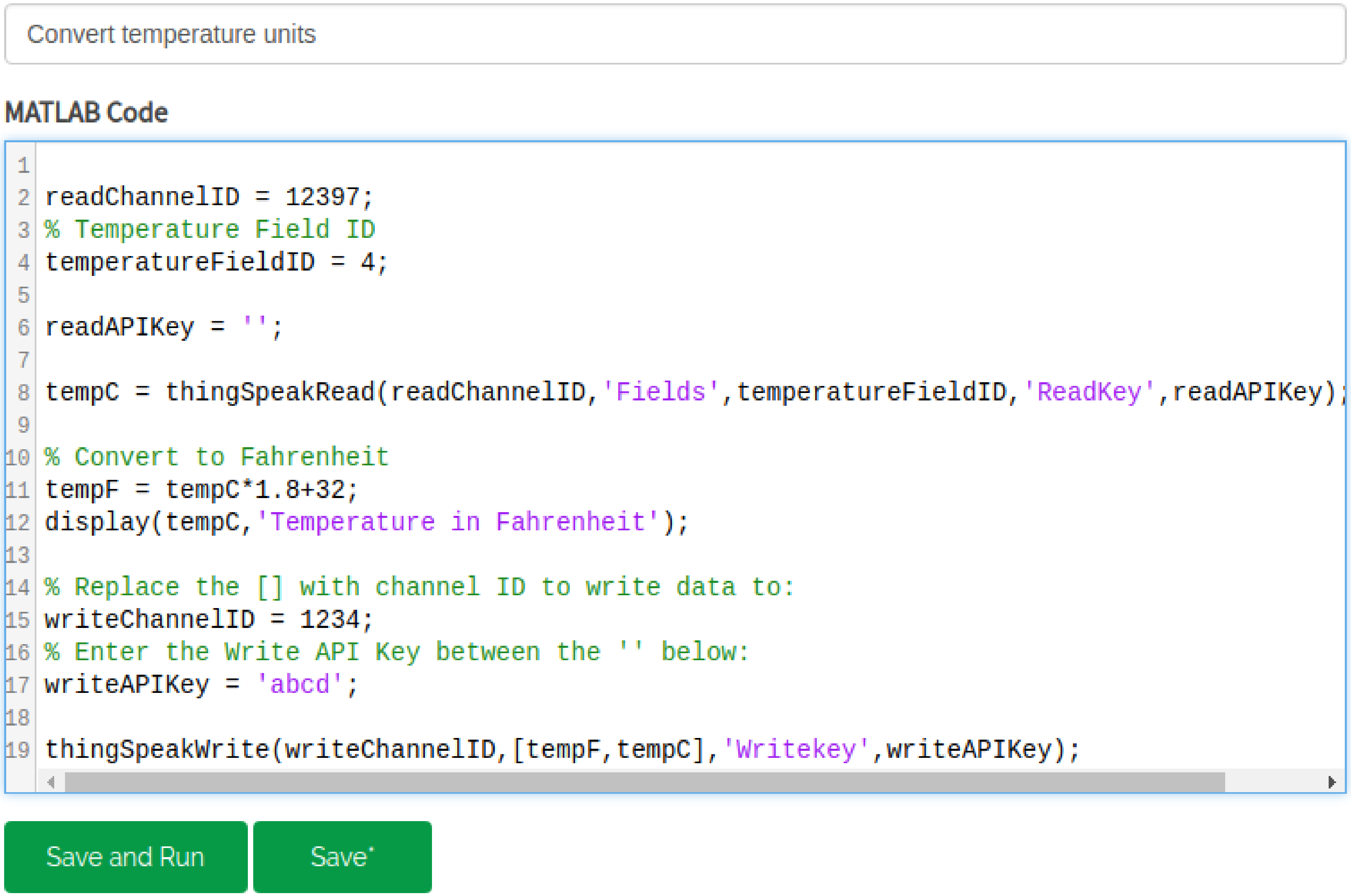
Apps / React / Fermenting Beer Too Cold

Name:	Fermenting Beer Too Cold
Condition Type:	Numeric
Test Frequency:	Every 30 minutes
Last Ran:	
Channel:	SensePi
Condition:	Field 1 (temperature) is less than 17
ThingTweet:	frankwalsh59: BEER TO COLD!!!
Run:	Only the first time the condition is met

ThingSpeak Example: Analysis

- Can write Matlab Code to analyse and transform data
- Possible uses:
  - Clean data (remove outliers)
  - Statistical analysis
  - Transfomations
  - Data Fusion
- Generally write results to second channel for further analysis/visualisation.

Thinkspeak: Convert Celcius to Fahrenheit



Convert temperature units

MATLAB Code

```

readChannelID = 12397;
% Temperature Field ID
temperatureFieldID = 4;

readAPIKey = '';

tempC = thingSpeakRead(readChannelID,'Fields',temperatureFieldID,'ReadKey',readAPIKey);

% Convert to Fahrenheit
tempF = tempC*1.8+32;
display(tempC,'Temperature in Fahrenheit');

% Replace the [] with channel ID to write data to:
writeChannelID = 1234;
% Enter the Write API Key between the '' below:
writeAPIKey = 'abcd';

thingSpeakWrite(writeChannelID,[tempF,tempC],'Writekey',writeAPIKey);
```

Comparison: Wia vs. Thingspeak

Category	Wia	Thingspeak
Device Programming/Interfacing	Several Curated SDKs for several device and languages. Simple publish: 3 lines of code (python)	Uses single purpose API (Rest interface). Use generic HTTP programming apstractions. Simple publish: 3 lines of code (pyhton)
Platform architecture	Device centric: Define device on platform "space" before conections. Events/commands based on device.	Channel Centric: Define a channel before connections. All apps and analysis use channels.
Data/Presistence model	Data contained in published "Events". Events need not be predefined.	Predefined channel fields(max 8). Devices write data to channel "scheme"
Data Visualisation/Dashboarding	Several "widgets" that can link to events including text/graph/image/map. HTML iFrames	Matlab charts. Maps/locations. HTML embedding iFrames

Comparison: Wia vs Thingspeak

Category	Wia	Thingspeak
3rd party Integrations	AWS recog., Slack, twitter, twillio,Sigfox,...	Twitter, Matlab/Simulink
Rules	Via Flows. Check values of state or fields (if temp<18 then command_heating_on	Use "React" apps to check for conditions (>,<,=..)
Analytics	Using Flows and custom functions in Javascript	Matlab charts and toolboxes. Lots od in-built example

Other Platforms

- Ubidots
- Amazon Web Services
- Microsoft Azure
- Evothings