IOTDB.org

Control all the Things with Node-JS

David Janes

@dpjanes
davidjanes@iotdb.org
http://iotdb.org/social/imadeit/

November 2014

Introduction

N.B.

- Demo example code on github
  - `dpjanes/`
  - `iotdb-examples/`
  - `demos/`
  - `2014-11-fsto/`

What is it?

```
{
    "on" : true,
    "temperature" : 225
}
```

What does each field mean?

Control or Measurement?

- Does on mean:
  - Turn it on? *or*
  - Is it on?
- Does temperature mean:
  - Set the temperature? *or*
  - What is the temperature?

Units

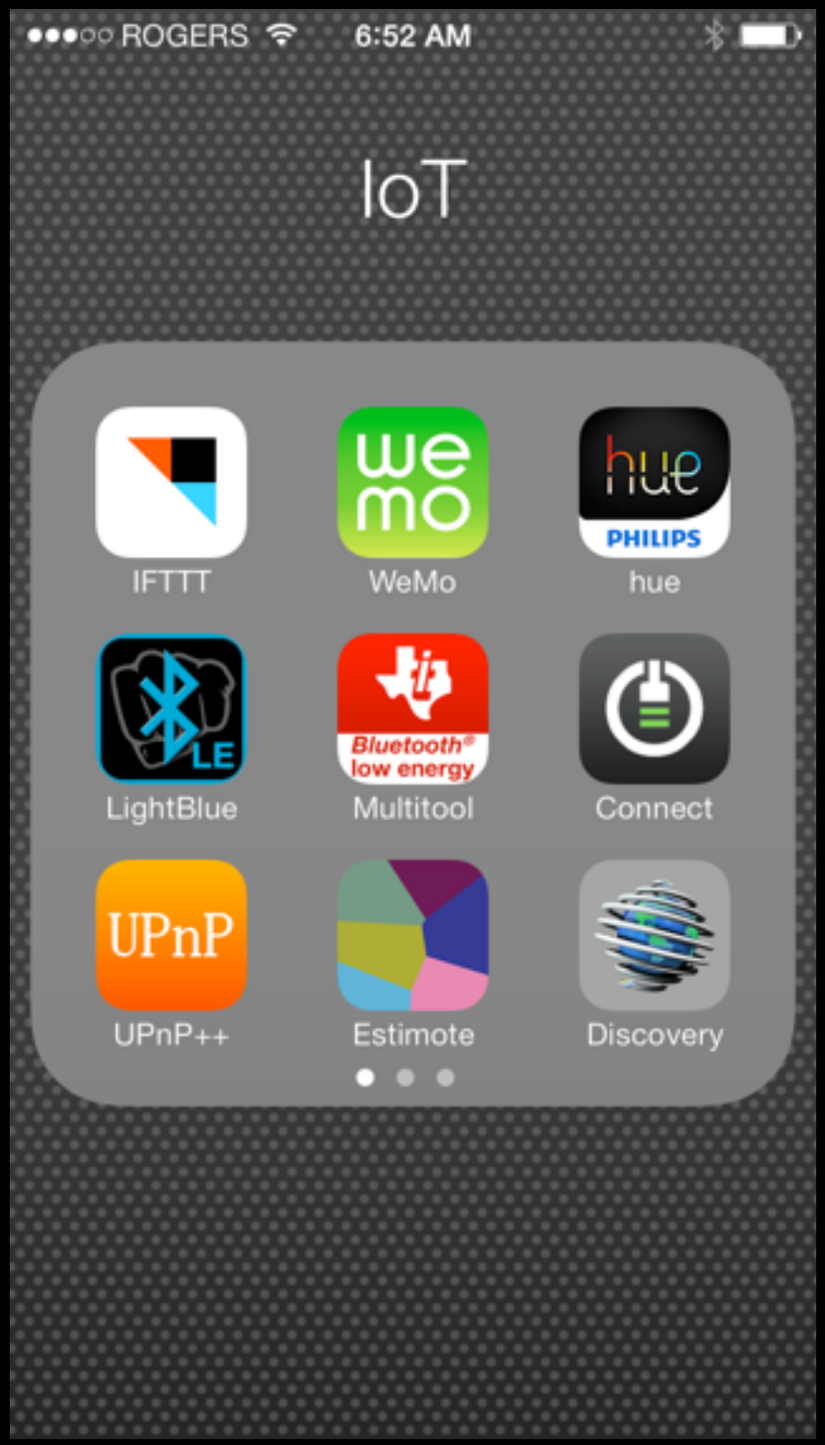
- What does temperature refer to?
  - degrees Celsius, Fahrenheit … or Kelvin?

What does the whole message represent?

What are we talking about … or with?

- An oven?
- A toaster?
- A sensor in the Large Hadron Collider?

Solve “the Basket Full of ~~Remotes~~ Apps Problem”


ROGERS
6:52 AM
IoT
IFTTT
WeMo
hue
PHILIPS
LE
LightBlue
Bluetooth®
low energy
Multitool
Connect
UPnP
UPnP++
Estimote
Discovery

Solve the N-standards

- XKCD comic here

IOTDB(.org)

Semantic Vocab

- Formal definitions
  - https://iotdb.org/pub
- JSON-LD

Models

- https://iotdb.org/iotdb
- https://github.com/dpjanes/iotdb-models

Node-IOTDB

IOTDB Stack

- Client Program
- Node-IOTDB
- Models
- Drivers
- Libraries

- `simple_on.js`
- `require('iotdb')`
- `WeMoSwitch`
- `iot-driver:upnp`
- `upnplib`

Client Program simple_red

```
iot
    .connect()
    .set(':color', 'red');
```

require('iotdb')

- locates Models, Drivers, Stores,…
- loads & maintains user configuration
- connects to iotdb.org (sometimes)
- manages Things (huge deal … too many things to go into in detail)

Model (I)

- Semantic description of Things
- (can be) written in JavaScript
- actually compiles to JSON-LD (with some restrained JavaScript)
- Node independent!

Model (II)

```
iotdb.make_model('WeMoSwitch')
    .facet(":device.switch")
    .product(“http://www.belkin.com/…”)
    .name("WeMo Switch")
    .description("Belkin WeMo Switch")
    .attribute(
        iotdb.make_boolean(":on")
            .name("on / off")
            .control()
    )
```

Model (III)

```
.driver_out(function(paramd) {
    if (paramd.thingd.on !== undefined) {
        paramd.driverd['urn:Belkin:service:basicevent:1'] = {
            'SetBinaryState' : {
                'BinaryState' : paramd.thingd.on ? 1 : 0
                }
            }
        }
    })
```

Drivers
iot-driver:upnp

- Binds Models to the code that actually "does the work"
- *discovery*
  - static configuration
  - dynamic / environmental

Libraries

- What developers usually program against
- We've rewritten a number of libraries to make them more reliable

Demos

simple_on

```
things = iot.connect()
things.set(':on', true);
```

simple_off

```
things = iot.connect()
things.set(':on', false);
```

simple_model

```
iot
    .connect(‘HueLight')
    .set(':on', true);
```

Metadata

Select Things

- select Things by (e.g.)
  - Model
  - name
  - number
  - place
  - facet

meta_dump

```
var things = iot.connect();

iot.on_things(function() {
    console.log(things.metas());
});
```

meta_model

```
iot.connect()
    .with_model("TCPConnectedLight")
    .set(':on', false)
```

meta_name

```
iot
.connect()
.with_name("Hue Lamp 2")
.set(':color', 'purple')
```

meta_number

```
iot
  .connect()
  .with_number(3)
  .set(':color', 'cyan')
```

meta_place

```
iot
  .connect()
  .with_room("David Bedroom")
  .with_floor("Second Floor")
  .set(':color', 'green')
```

meta_facet

```
iot
    .connect()
    .with_facet(":device.lighting")
    .set(':on', true);
```

where does Metadata come from?

- Metadata comes from several places
  - can be altered in code
  - can be persisted to locally / to disk
  - can be retrieved from iotdb.org
  - “inherent” in the Model / Driver

Events

two types of events

- `thing.on(key, callback)`
- `thing.on_change(callback)`

event_brightness

```
var lights = iot.connect()
var input = iot.connect({
    model: "FirmataInputUnit",
    pin: 0
});
input.on(":value", function(thing, attribute, value) {
    lights.set(':brightness', value);
})
```

event_fob

```
var things = iot.connect();

iot
    .connect('TIKeyFob')
    .on('left', function() {
        things.set(':on', true);
    })
    .on('right', function() {
        things.set(':on', false);
    })
```

Arduino / Firmata

firmata_cycle

```
var things = iot
    .connect()
    .connect({
        model: "FirmataNeoPixel",
        pin: 6,
        n: 16
    })

var colors = [ "red", "green", "blue", "white", "black" ];
var ci = 0;

setInterval(function() {
    things.set(":color", colors[ci++ % colors.length]);
}, 2500)
```

firmata_neopixel

```
var n = 16;
var leds = iot.connect({
    model: "FirmataNeoPixel",
    pin: 6,
    n: n
})

var c = new iotdb.libs.Color()
var ci = 0;
var cf = 0;

setInterval(function() {
    c.set_hsl(cf, 1, 0.5)
    cf += 0.015;
    if (cf > 1) cf = 0;

    leds
        .with_number(ci++ % n)
        .set(":color", c.get_hex())
        ;

}, 50)
```

Arduino Models

- FirmataChainableLED
- FirmataDHT11
- FirmataGroveThermistor
- FirmataInputBoolean
- FirmataInputUnit
- FirmataLightDimmer
- FirmataLightSensor
- FirmataLightSimple
- FirmataMotionSensor
- FirmataNeoPixel
- FirmataOn
- FirmataOutputBoolean

Stores

Available Stores

- dweet
- http
- mqtt
- phant
- pubnub
- thingspeak

store_phant

```
var input = iot.connect('TIKeyFob')
```

```
var store = iot
    .store('phant')
    .track(input)
```

https://data.sparkfun.com/streams/aGNjQK9RwZHgEmx089Lg

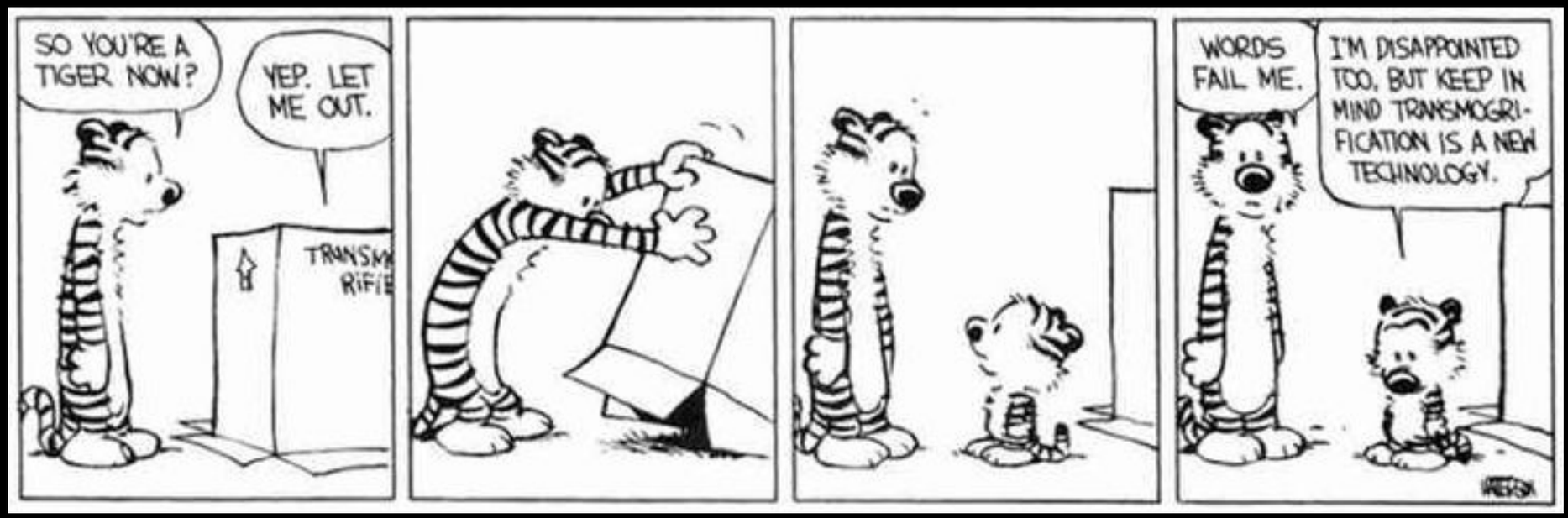
store_mqtt

```
var input = iot.connect({
    model: "FirmataInputUnit",
    pin: 0
});

var store = iot
    .store('mqtt')
    .track(input)
```

mqtt.iotdb.org u/dpjanes/#

Transmogrifiers


SO YOU'RE A TIGER NOW?
YEP. LET ME OUT.
TRANSM
RIFIE
WORDS FAIL ME.
I'M DISAPPOINTED TOO, BUT KEEP IN MIND TRANSMOGRI-FICATION IS A NEW TECHNOLOGY.

What if we have the wrong thing?

- Have Celsius but want Fahrenheit
- Want to set brightness but only have color
- What if brightness is 0-100 on one device and 0-1 on another
- Average data? Max data over time? &c…

trans_fahrenheit

```
var t_c = iot.connect("FirmataDHT11")
t_c.on('temperature', function(thing, attribute, value) {
    console.log("+ temperature (C)", value)
})
```

```
var t_f = t_c.transmogrify(
  iot.transmogrifier(":imperial/fahrenheit"))
t_f.on('temperature', function(thing, attribute, value) {
    console.log("+ temperature (F)", value)
})
```

warning

- transmogrifiers are very much a work in progress!
- not even pushed to npm yet

Takeaways

IOTDB

- Semantic description of how things work

Node-IOTDB

- Powerful language for controlling things
- Builds on IOTDB concepts
- Strong alpha
- Join! Help!

Modeling

```
{
    "@context": "./meta-iri.jsonld",
    "on": true,
    "temperature": 225
}
```

Note: @context is just one way of doing this!

N.B.

- The data does not have to be JSON!
- You don’t need to have IOTDB!
- You don’t need to use IOTDB IRIs (but you should)
  - https://iotdb.org/pub/

Additional Resources

- Many examples + House of Janes
  https://github.com/dpjanes/iotdb-examples
- Getting started
  https://iotdb.org/docs/node/getting-started
- Concepts
  https://medium.com/@dpjanes

One last point

- Don’t be misled by the simplicity!
- I worked backward from “what do I want to do” to get the code
- Way more things than I demoed today
- I want to work with you!

Get in touch!

David Janes

@dpjanes
davidjanes@iotdb.org
http://iotdb.org/social/imadeit/