



# oneM2M Dashboard

Yungi Park (yun0429@keti.re.kr)

Korea Electronics Technology Institute

2020.11.13

The project "International Digital Cooperation - ICT Standardisation" is funded by the European Union



- Prerequisites
- oneM2M – Dashboard
  - Dashboard Overview
  - Dashboard Scenario
  - Dashboard Installation
  - oneM2M - Dashboard Interworking

# Prerequisites

- S/W
  - Node.js
  - Visual Studio Code
    - <https://code.visualstudio.com/download>
  - cmdr(Terminal)
    - <https://cmdr.net/>
  - Dashboard Source
    - <https://github.com/Freeboard/freeboard.git>



# oneM2M - Dashboard

# Dashboard Overview

- Freeboard is the free IoT dashboard based on a turn-key HTML engine
  - supports REST API with JSON serialization and HTTP extension headers which is necessary for oneM2M
- Freeboard provides a plugin architecture for creating data sources (which fetching data) and widgets (which displaying data)



# Dashboard Scenario



- Firstly, generate the data using the Device Simulator (Day 1)
- Secondly, connect the freeboard and oneM2M platform using oneM2M API
- Then you can monitor the data of each device through various widgets the dashboard

# Dashboard Installation

- Install required node modules
  - Install the required node modules using the following command:
    - > npm install
    - > grunt
      - > npm install -g grunt-cli
      - > grunt

```
C:\Users\Parkyun\Desktop\freeboard-master
> npm install
npm WARN freeboard@1.1.3 No repository field.
npm WARN freeboard@1.1.3 No license field.

added 146 packages from 177 contributors and audited 148 packages in 10.5s

2 packages are looking for funding
  run `npm fund` for details

found 30 vulnerabilities (10 low, 3 moderate, 17 high)
  run `npm audit fix` to fix them, or `npm audit` for details
```

```
C:\Users\Parkyun\Desktop\freeboard-master
> grunt
Running "concat:css" (concat) task
File "css/freeboard.css" created.

Running "cssmin:css" (cssmin) task
File "css/freeboard.min.css" created.

Running "concat:fb" (concat) task
File "js/freeboard.js" created.

Running "concat:thirdparty" (concat) task
File "js/freeboard.thirdparty.js" created.

Running "concat:plugins" (concat) task
File "js/freeboard.plugins.js" created.

Running "concat:fb_plugins" (concat) task
File "js/freeboard_plugins.js" created.

Running "uglify:fb" (uglify) task
>> 1 file created.

Running "uglify:plugins" (uglify) task
>> 1 file created.

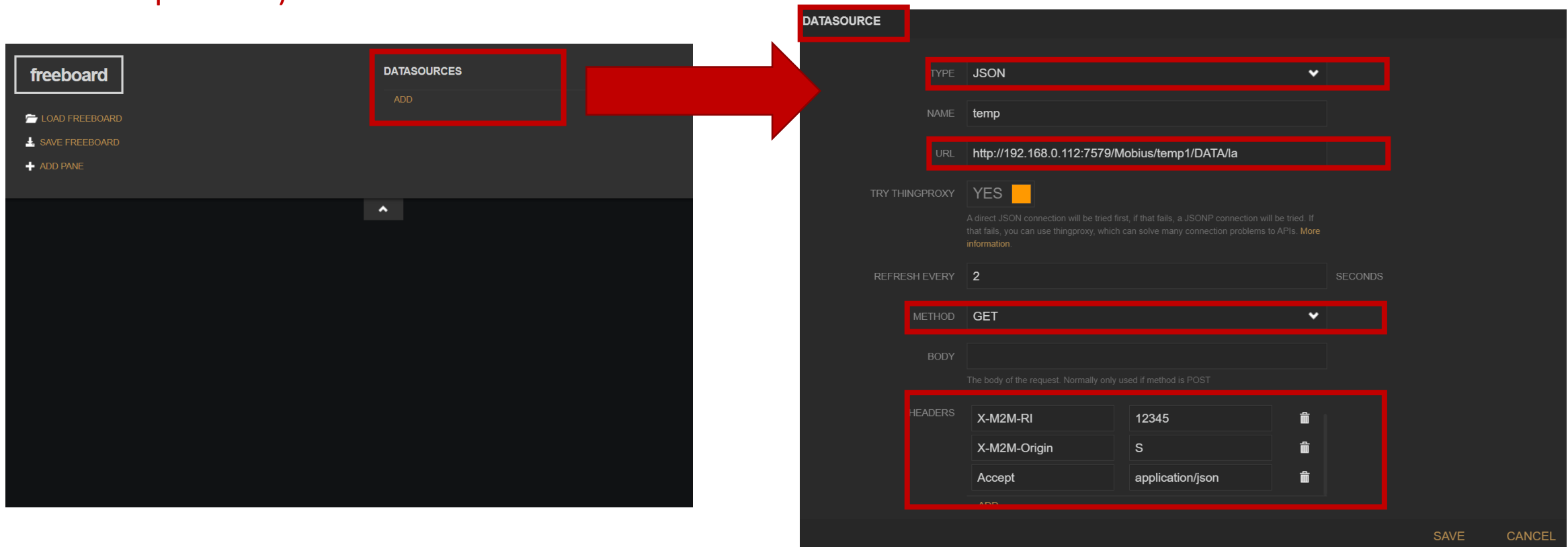
Running "uglify:fb_plugins" (uglify) task
>> 1 file created.

Running "uglify:thirdparty" (uglify) task
>> 1 file created.

Running "string-replace:css" (string-replace) task
File "css/freeboard.css" created.
File "css/freeboard.min.css" created.
```

# Dashboard Setup

- Create data source
  - For URL, you must enter the IP of oneM2M platform (\*do not use 127.0.0.1 due to CORS problem)



The image shows a two-part screenshot of a dashboard setup. The left part shows the 'freeboard' interface with a 'DATASOURCES' button highlighted by a red box. A large red arrow points from this button to the right part of the image, which shows the 'DATASOURCE' configuration form. The form has several fields highlighted with red boxes: 'TYPE' set to 'JSON', 'NAME' set to 'temp', 'URL' set to 'http://192.168.0.112:7579/Mobius/temp1/DATA/la', 'TRY THINGPROXY' set to 'YES', 'REFRESH EVERY' set to '2' seconds, 'METHOD' set to 'GET', and a 'HEADERS' table with three entries: 'X-M2M-Rl' with value '12345', 'X-M2M-Origin' with value 'S', and 'Accept' with value 'application/json'. The 'SAVE' and 'CANCEL' buttons are at the bottom right of the form.

**freeboard**

- LOAD FREEBOARD
- SAVE FREEBOARD
- + ADD PANE

**DATASOURCES**

ADD

**DATASOURCE**

TYPE: JSON

NAME: temp

URL: http://192.168.0.112:7579/Mobius/temp1/DATA/la

TRY THINGPROXY: YES

REFRESH EVERY: 2 SECONDS

METHOD: GET

BODY

HEADERS

X-M2M-Rl	12345	
X-M2M-Origin	S	
Accept	application/json	

SAVE CANCEL



# Dashboard Setup

- Add panes to have your own widgets



# Dashboard Setup

- Add text widget

WIDGET

TYPE

TITLE

SIZE

VALUE  + DATASOURCE

INCLUDE SPARKLINE ☐ NO

ANIMATE VALUE CHANGES ☒ YES

UNITS



WIDGET

TYPE

TITLE

SIZE

VALUE  + DATASOURCE JS EDITOR

INCLUDE SPARKLINE ☐ NO

ANIMATE VALUE CHANGES ☒ YES

UNITS

SAVE CANCEL

# Dashboard Setup

- Add sparkline widget

WIDGET

TYPE

TITLE

VALUE  + DATASOURCE

+ ADD

INCLUDE LEGEND ☐ NO

LEGEND

Comma-separated for multiple sparklines



TYPE

TITLE

VALUE

+ ADD

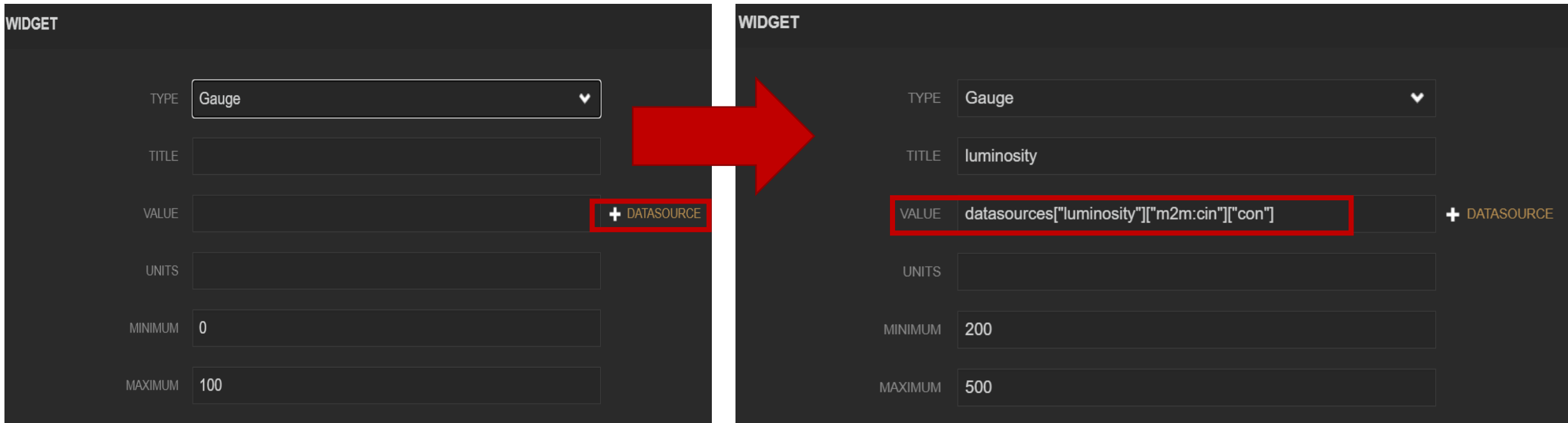
INCLUDE LEGEND ☐ NO

LEGEND

Comma-separated for multiple sparklines

# Dashboard Setup

- Add gauge widget



The diagram illustrates the configuration of a gauge widget in two steps, connected by a large red arrow.

**Initial Configuration (Left):**

- WIDGET**
- TYPE:** Gauge
- TITLE:** (empty)
- VALUE:** (empty) with a **+ DATASOURCE** button next to it.
- UNITS:** (empty)
- MINIMUM:** 0
- MAXIMUM:** 100

**Final Configuration (Right):**

- WIDGET**
- TYPE:** Gauge
- TITLE:** luminosity
- VALUE:** `datasources["luminosity"]["m2m:cin"]["con"]` (highlighted with a red box) with a **+ DATASOURCE** button next to it.
- UNITS:** (empty)
- MINIMUM:** 200
- MAXIMUM:** 500

# Dashboard Setup

- Add indicator light widget
  - Have your own code which returns either '0' or '10'

```
var data = datasources["lamp"]["m2m:cin"]["con"];  
  
if(data ="1"){  
    return 1  
}  
else {  
    return 0  
}
```

WIDGET

TYPE: Indicator Light

TITLE:

VALUE: |

+ DATASOURCE

JS EDITOR

ON TEXT: + DATASOURCE JS EDITOR

OFF TEXT: + DATASOURCE JS EDITOR

SAVE CA

WIDGET

Indicator Light

TITLE:

VALUE: var data = datasources["lamp"]["m2m:cin"]["con"];

+ DATASOURCE JS EDITOR

ON TEXT: LAMP ON + DATASOURCE JS EDITOR

OFF TEXT: LAMP OFF + DATASOURCE JS EDITOR

# Practice

- Configure your dashboard like this
- And then provide data with Device Simulator
- See what happens on the dashboard
- How can you use this simple dashboard for your application?

