

# AEGIS 系統簡介及安裝

---

## 系統簡介

這是一個包含維運的電力管理系統

## 相關服務

此系統需搭配 postgres, influxdb, nodered 等背景服務才能運作。此處以 Docker 方式在背景服務，詳細定義在 "docker-compose.yml"，可用下方指令開關背景服務

### 啟動背景服務

```
./start_docker.sh
```

### 停止背景服務

```
./stop_docker.sh
```

## AEGIS 安裝

### 安裝 nodejs

此程式由 nodejs 寫成，請先到 nodejs 官網下載並安裝

### 安裝相關套件

```
$ npm install
```

在 macOS 可能出現 canvas 無法安裝的問題 (若無問題可跳過)

### 安裝 node-gyp

```
sudo npm install -g npm  
sudo npm install -g node-gyp
```

### 安裝系統套件

```
brew install pkg-config cairo pango libpng jpeg giflib librsvg  
npm install canvas
```

### 若還是有問題,從原始碼編譯 (沒問題可跳過)

```
npm install canvas --build-from-source
```

## 執行

```
# development
$ npm run start

# watch mode
$ npm run start:dev

# production mode
$ npm run start:prod
```

## API 測試網頁

- Restful API - <http://127.0.0.1:3000/api>
- GraphQL - <http://127.0.0.1:3000/graphql>

## 登入系統 - POST "/auth/login"

- API說明：
  - 用來登入系統
- HTTP Header：
  - Content-Type: application/json
- 欄位說明：
  - user\_account: 帳號
  - user\_password: 密碼

欄位範例：

```
{
  "user_account": "admin",
  "user_password": "xxx"
}
```

傳回結果：

```
{
  "access_token": "eyJ23.xx",
  "user_id": 2,
  "user_name": "Accton Admin",
  "role_id": 2,
  "domain_id": 2,
  "domain_name": "taipei",
  "organization_id": 2
}
```

## Restful API 測試

- GET "/api/domains"
  - HTTP Header :
    - Authorization: login 傳回來的 access\_token ◦ 例如: "Bearer ey.123.xx"
- 傳回

```
[
  {
    "domain_id": 48,
    "domain_name": "Test-domain",
    "parent_domain_id": 2,
    "is_organization": false,
    "address": "",
    "lat": "23.75518177",
    "lng": "120.80566406",
    "zoom": 7,
    "photos": "",
    "feed_in_tariffs": null,
    "feed_in_tariff_now": null
  }
]
```

## GraphQL 測試

### 取得 device 狀態

- 用瀏覽器打開 <http://127.0.0.1:3000/graphql>
- 左方輸入

```
query {
  device(device_id:1) {
    device_state
  }
}
```

- 按中央的三角形查詢傳回下面結果

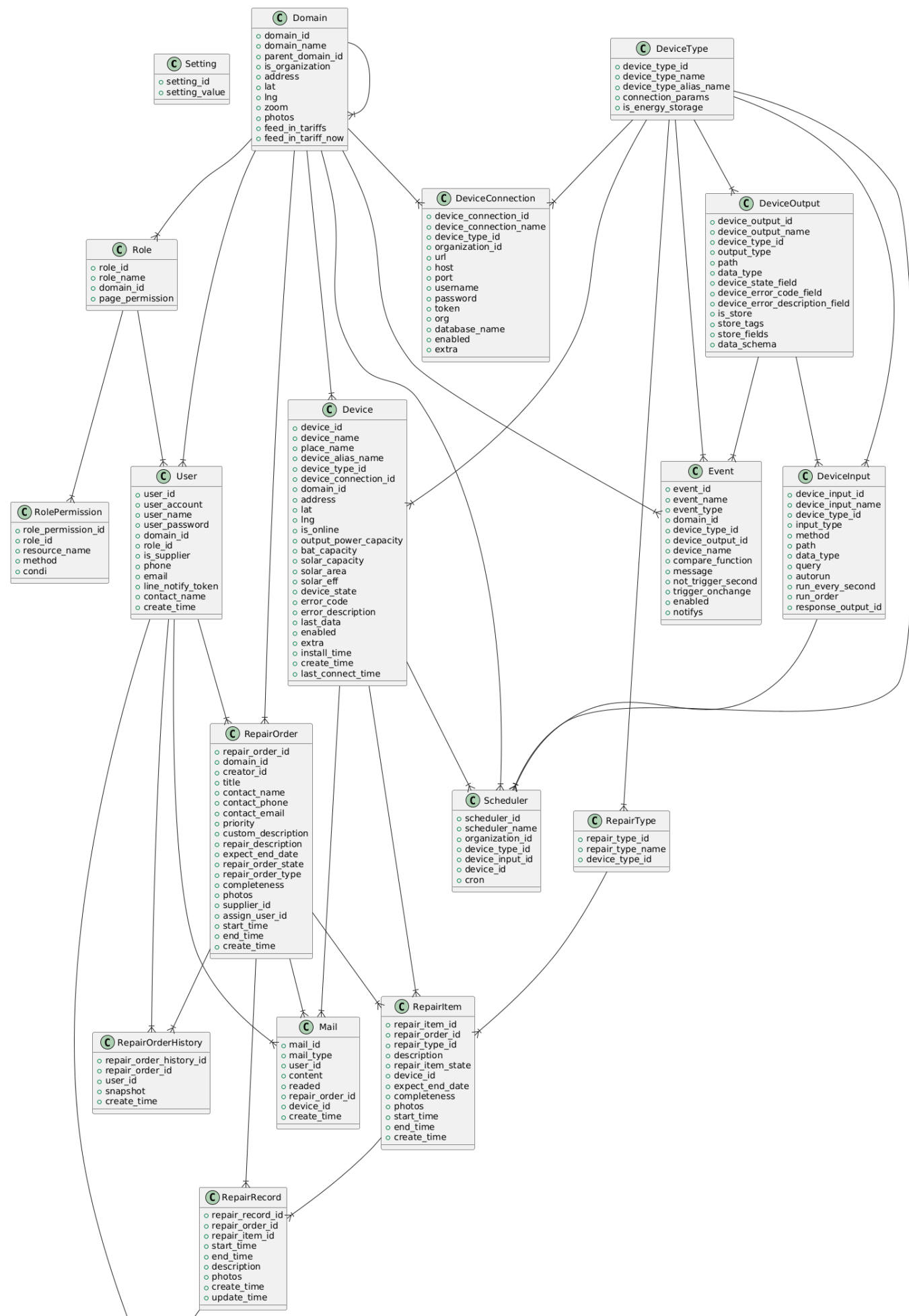
```
"data": {
  "device": {
    "device_state": "running"
  }
}
```

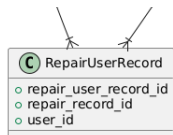
# 資料庫

## 資料表定義

Name	Description
Setting	設置
Domain	域
Role	角色
Role Permission	角色權限
User	用戶
Device Type	裝置類型
Device Connection	裝置連線
Device Output	裝置輸出
Device Input	裝置輸入
Device	裝置
Scheduler	排程器
Repair Type	維修類型
Repair Order	維修訂單
Repair Item	維修項目
Repair Order History	維修訂單歷史
Repair Record	維修記錄
Repair User Record	維修派工記錄
Mail	郵件
Event	事件

# 資料庫關聯圖





## Setting (設置)

### 說明

用來儲存設置的鍵值對

### 欄位定義

Name	Type	Description
setting_id	VARCHAR(127)	設置ID
setting_value	TEXT	設置值

## Domain (域)

### 說明

定義系統中的域，支持層級結構

### 欄位定義

Name	Type	Description
domain_id	SERIAL	域ID
domain_name	VARCHAR(64)	域名稱
parent_domain_id	INT	父域ID
is_organization	BOOLEAN	是否為組織
address	VARCHAR(255)	地址
lat	DECIMAL(10,8)	緯度
lng	DECIMAL(11,8)	經度
zoom	FLOAT	地圖縮放級別
photos	TEXT	照片
feed_in_tariffs	TEXT	躉售費率
feed_in_tariff_now	FLOAT	當前躉售費率

### 其它 API

- PUT /api/domains/{domain\_id}/replace\_photo
  - 說明: domain 照片上傳
  - 參數:
    - domain\_id: 域ID
    - multipart/form-data
      - file: 要上傳的照片

## Role (角色)

說明

定義角色和其所屬域的權限

欄位定義

Name	Type	Description
role_id	SERIAL	角色ID
role_name	VARCHAR(64)	角色名稱
domain_id	INT	所屬域ID
page_permission	TEXT	頁面權限

## Role Permission (角色權限) (目前未實做)

說明

角色與資源的權限對應

欄位定義

Name	Type	Description
role_permission_id	SERIAL	角色權限ID
role_id	INT	角色ID
resource_name	VARCHAR(127)	資源名稱
method	VARCHAR(31)	方法
condi	VARCHAR(255)	條件

## User (用戶)

說明

系統中的用戶信息

欄位定義

Name	Type	Description
user_id	SERIAL	用戶ID
user_account	VARCHAR(64)	用戶帳號
user_name	VARCHAR(64)	用戶名稱
user_password	VARCHAR(64)	用戶密碼
domain_id	INT	所屬域ID
role_id	INT	角色ID

Name	Type	Description
is_supplier	BOOLEAN	是否為供應商
phone	VARCHAR(32)	電話
email	VARCHAR(63)	電子郵件
line_notify_token	VARCHAR(127)	Line 通知令牌
contact_name	VARCHAR(64)	聯絡人名稱
create_time	TIMESTAMP	創建時間

## Device Type (裝置類型)

### 說明

用來定義裝置的類型

### 欄位定義

Name	Type	Description
device_type_id	SERIAL	裝置類型ID
device_type_name	VARCHAR(255)	裝置類型名稱
device_type_alias_name	VARCHAR(255)	裝置類型別名
connection_params	TEXT	連線時需要的參數
is_energy_storage	BOOLEAN	是否為儲能裝置

## Device Connection (裝置連線)

### 說明

定義裝置的連線配置, 例如 mqtt 或 mysql 等需要常連線的連線方式必須使用

### 欄位定義

Name	Type	Description
device_connection_id	SERIAL	裝置連線ID
device_connection_name	VARCHAR(127)	裝置連線名稱
device_type_id	INT	裝置類型ID
organization_id	INT	組織ID
url	TEXT	連接URL
host	VARCHAR(255)	主機
port	INT	埠號
username	VARCHAR(127)	使用者名稱
password	VARCHAR(127)	密碼



Name	Type	Description
token	VARCHAR(255)	令牌
org	VARCHAR(127)	組織
database_name	VARCHAR(63)	數據庫名稱
enabled	BOOLEAN	是否啟用
extra	TEXT	額外信息

## Device Output (裝置輸出)

### 說明

定義裝置的輸出信息

### 欄位定義

Name	Type	Description
device_output_id	SERIAL	裝置輸出ID
device_output_name	VARCHAR(127)	裝置輸出名稱
device_type_id	INT	裝置類型ID
output_type	VARCHAR(63)	輸出類型, ex: one of {"mqtt", "modbus", "http"}
path	TEXT	路徑。 <div>當 "output_type" 為下面值的範例： modbus: 用來指定 modbus device_id. ex:"1" mqtt: 用來指定 mqtt topic. ex: "/path/to/topic1" http: 用來指定 http url. ex: "http://127.0.0.1:3000/test.html"</div>
data_type	VARCHAR(31)	數據類型 ex: one of {"text", "json"}
device_state_field	VARCHAR(63)	更新 device_state 的欄位
device_error_code_field	VARCHAR(63)	更新 device_error_code 的欄位
device_error_description_field	TEXT	更新 device_error_description 的欄位
is_store	BOOLEAN	是否儲存
store_tags	TEXT	<div>influxdb tag ex:[ {"tag":"place_name","source":"device_order"}, {"tag":"device_name","source":"device_name"}, {"tag":"device_sn","source":"device_sn"} ]</div>

Name	Type	Description
store_fields	TEXT	<pre>influxdb field ex:[ {"field":"LoadPower","source":"LoadPower"}, {"field":"SOC","source":"SOC"} ]</pre>
data_schema	TEXT	數據模式

## data\_schema

data\_schema 須符合 "json schema" 的規範, 請參照下面網址:

- JSON Schema: <https://json-schema.org/>
- JSON Schema Builder: <https://bjdash.github.io/JSON-Schema-Builder/>
- JSON Schema validator: <https://github.com/ajv-validator/ajv>

下面範例中 "productId" 和 "productName" 為欄位名稱，型態分別為整數和字串。

```
{
  "type": "object",
  "properties": {
    "productId": {
      "description": "The unique identifier for a product",
      "type": "integer"
    },
    "productName": {
      "description": "Name of the product",
      "type": "string"
    },
    "createdDate": {
      "type": "string",
      "format": "date"
    },
    "updatedAtTime": {
      "type": "string",
      "format": "date-time"
    }
  }
}
```

```
{
  "type": "array",
  "items": {
    "type": "string"
  },
  "minItems": 1
}
```

# Device Input (裝置輸入) (目前未實做)

## 說明

定義裝置的輸入信息

## 欄位定義

Name	Type	Description
device_input_id	SERIAL	裝置輸入ID
device_input_name	VARCHAR(127)	裝置輸入名稱
device_type_id	INT	裝置類型ID
input_type	VARCHAR(63)	輸入類型 ex: one of {"mqtt", "modbus", "http", "mysql", "influxdb"}
method	VARCHAR(63)	存取方式 ex: one of {"get", "post", "patch", "put", "delete"}
path	TEXT	路徑
data_type	VARCHAR(31)	數據類型 ex: one of {"text", "json"}
query	TEXT	查詢語法
autorun	BOOLEAN	是否自動運行
run_every_second	INT	每秒運行
run_order	INT	運行順序
response_output_id	INT	當 "input_type" 使用 "http" 時, 觸發的回應 output 處理方式

## query 查詢語法說明

"query" 及 "path" 欄位可用 "\${xxx}" 來替換變數, 目前可用的項目如下：

- \${param.xxx} 會被替換成 "/device\_types/{device\_type\_id}/trigger" API 中的 "param.xxx" 欄位
- \${device.device\_name} 會被替換成 "/device\_types/{device\_type\_id}/trigger" API 中, "domain\_id" 及 "device\_name" 所包含的裝置名稱

當 "input\_type" 為下面值的範例:

```
modbus:
  path:
    1
  query:
    123

mqtt:
  path:
    /devices/${device.device_name}/control
  query:
    data_type=="text"
      ${param}
    data_type=="json"
      {const1:'const1', var1:${param.var1}, all_data:${data}}
```

http:

```
method:
  post
path:
  /devices/${device.device_name}/control
query:
  data_type=="text"
    ${param}
  data_type=="json"
    {const1:'const1', var1:${param.var1}, all_data:${data}}

mysql:
  query:
    SELECT * FROM user

influxdb:
  query:
    from(bucket: "bucket1") \> range(start: 0, stop: 100) \> filter(fn: (r) =>
r["_measurement"] == "measurement1")
```

## Device (裝置)

### 說明

定義裝置的信息

### 欄位定義

Name	Type	Description
device_id	SERIAL	裝置ID
device_name	VARCHAR(127)	裝置名稱
place_name	VARCHAR(127)	地點名稱
device_alias_name	VARCHAR(127)	裝置別名
device_type_id	INT	裝置類型ID
device_connection_id	INT	裝置連線ID
domain_id	INT	所屬域ID
address	VARCHAR(255)	地址
lat	DECIMAL(10,8)	緯度
lng	DECIMAL(11,8)	經度
is_online	BOOLEAN	是否在線
output_power_capacity	FLOAT	輸出功率容量
bat_capacity	FLOAT	電池容量
solar_capacity	FLOAT	太陽能容量
solar_area	FLOAT	太陽能面積
solar_eff	FLOAT	太陽能效率

Name	Type	Description
device_state	VARCHAR(31)	裝置狀態
error_code	VARCHAR(63)	錯誤代碼
error_description	TEXT	錯誤描述
last_data	TEXT	最後數據
enabled	BOOLEAN	是否啟用
extra	TEXT	額外信息
install_time	TIMESTAMP	安裝時間
create_time	TIMESTAMP	創建時間
last_connect_time	TIMESTAMP	最後連接時間

device\_state

- 正常運作中 running
- 警告 warning
- 錯誤 error

Scheduler (排程器) (目前未實做)

說明

定義排程器的排程信息

欄位定義

Name	Type	Description
scheduler_id	SERIAL	排程器ID
scheduler_name	VARCHAR(127)	排程器名稱
organization_id	INT	組織ID
device_type_id	INT	裝置類型ID
device_input_id	INT	要觸發的裝置輸入ID
device_id	INT	裝置ID
cron	VARCHAR(63)	CRON 表達式

cron 設定範例

```
# 每天早上 8 點 30 分執行
30 08 * * *

# 每週日下午 6 點 30 分執行
30 18 * * 0

# 每週日下午 6 點 30 分執行
30 18 * * Sun
```

```
# 每年 6 月 10 日早上 8 點 30 分執行
30 08 10 06 *

# 每月 1 日、15 日、29 日晚上 9 點 30 分各執行一次
30 21 1,15,29 * *

# 每隔 10 分鐘執行一次
*/10 * * * *

# 從早上 9 點到下午 6 點，凡遇到整點就執行
00 09-18 * * *
```

## Repair Type (維修類型)

### 說明

### 定義維修類型

### 欄位定義

Name	Type	Description
repair_type_id	SERIAL	維修類型ID
repair_type_name	VARCHAR(255)	維修類型名稱
device_type_id	INT	裝置類型ID

### 傳送資料給裝置

POST /device\_types/{device\_type\_id}/trigger

```
{
  "domain_id": 3, // (選填) 傳送給此 domain 下所有裝置
  "device_name": "device1", // (選填) 傳送單一裝置
  "device_input_id": 1, // (必填) 裝置的哪個輸入端
  "param": {"temp":12} // (選填) 帶入參數，可將 "DeviceInput" 中, "path" 及 "query" 中的 ${param.temp} 替換成 12
}
```

## Repair Order (維修訂單)

### 說明

### 維修訂單的詳細信息

### 欄位定義

Name	Type	Description
repair_order_id	SERIAL	維修訂單ID
domain_id	INT	所屬域ID

Name	Type	Description
creator_id	INT	創建者ID
title	VARCHAR(127)	標題
contact_name	VARCHAR(63)	聯絡人名稱
contact_phone	VARCHAR(31)	聯絡電話
contact_email	VARCHAR(127)	聯絡郵件
priority	INT	優先級
custom_description	TEXT	自定義描述
repair_description	TEXT	維修描述
expect_end_date	DATE	預期完成日期
repair_order_state	VARCHAR(31)	維修訂單狀態
repair_order_type	VARCHAR(31)	維修訂單類型
completeness	FLOAT	完整性
photos	TEXT	照片
supplier_id	INT	供應商ID
assign_user_id	INT	指派用戶ID
start_time	TIMESTAMP	開始時間
end_time	TIMESTAMP	結束時間
create_time	TIMESTAMP	創建時間

repair\_order\_state, repair\_order\_type

- 故障回報單 report\_order
  - 待處理 pending
  - 已拒絕 rejected
- 維修單 repair\_order
  - 未派工 not\_dispatched
  - 待處理 pending
  - 維修中 repairing
  - 維修完成 completed
- 已結案 closed
  - 正常結案 normal
  - 異常結案 abnormal

其它 API

- PUT /api/repair\_orders/{repair\_order\_id}/replace\_photo
  - 說明: repair\_order 照片上傳
  - 參數:

- repair\_order\_id: repair\_order ID
- multipart/form-data
  - file: 要上傳的照片

## Repair Item (維修項目)

說明

維修訂單中的維修項目

欄位定義

Name	Type	Description
repair_item_id	SERIAL	維修項目ID
repair_order_id	INT	維修訂單ID
repair_type_id	INT	維修類型ID
description	TEXT	描述
repair_item_state	VARCHAR(31)	維修項目狀態
device_id	INT	裝置ID
expect_end_date	DATE	預期完成日期
completeness	FLOAT	完整性
photos	TEXT	照片
start_time	TIMESTAMP	開始時間
end_time	TIMESTAMP	結束時間
create_time	TIMESTAMP	創建時間

## Repair Order History (維修訂單歷史)

說明

維修訂單的歷史紀錄

欄位定義

Name	Type	Description
repair_order_history_id	SERIAL	維修訂單歷史ID
repair_order_id	INT	維修訂單ID
user_id	INT	用戶ID
snapshot	TEXT	快照
create_time	TIMESTAMP	創建時間

## Repair Record (維修記錄)

說明



維修記錄的詳細信息

欄位定義

Name	Type	Description
repair_record_id	SERIAL	維修記錄ID
repair_order_id	INT	維修訂單ID
repair_item_id	INT	維修項目ID
start_time	TIMESTAMP	開始時間
end_time	TIMESTAMP	結束時間
description	TEXT	描述
photos	TEXT	照片
create_time	TIMESTAMP	創建時間
update_time	TIMESTAMP	更新時間

其它 API

- PUT /api/repair\_records/{repair\_record\_id}/replace\_photo
  - 說明: repair\_record 照片上傳
  - 參數:
    - repair\_record\_id: repair\_record ID
    - multipart/form-data
      - file: 要上傳的照片

Repair User Record (維修派工記錄)

說明

維修派工記錄

欄位定義

Name	Type	Description
repair_user_record_id	SERIAL	派工記錄ID
repair_record_id	INT	維修記錄ID
user_id	INT	用戶ID

Mail (郵件)

說明

用於管理郵件通知

欄位定義

Name	Type	Description
------	------	-------------

Name	Type	Description
mail_id	SERIAL	郵件ID
mail_type	VARCHAR(63)	郵件類型
user_id	INT	用戶ID
content	TEXT	內容
readed	BOOLEAN	是否已讀
repair_order_id	INT	維修訂單ID
device_id	INT	裝置ID
create_time	TIMESTAMP	創建時間

### 其它 API

- DELETE /api/emails/multi/{ids}
  - 說明: 一次刪除多個 mail
  - 參數:
    - ids: 要刪除的 mail\_id, 以 "," 隔開。例如 : "12,34,56"

## Event (觸發事件的規則)

### 說明

### 觸發事件的規則

### 欄位定義

Name	Type	Description
event_id	SERIAL	事件ID
event_name	VARCHAR(63)	事件名稱
event_type	VARCHAR(63)	事件類型, 可以是下面三種 {"user": "使用者觸發", "device": "裝置觸發", "repair_order": "維修單觸發"}
domain_id	INT	所屬域ID
device_type_id	INT	裝置類型ID (for event_type="device")
device_output_id	INT	裝置輸出ID
device_name	VARCHAR(127)	裝置名稱 (for event_type="device")
compare_function	TEXT	比較函數, 必須傳回true或false, 傳回true時觸發事件 (for event_type="device")
message	TEXT	訊息
not_trigger_second	INT	單一裝置幾秒內不要重複觸發 (for event_type="device")
trigger_onchange	BOOLEAN	是否觸發變更
enabled	BOOLEAN	是否啟用
notifys	TEXT	要通知的對象

## notifys 範例

```
[
  {
    "user_id": 35,
    "mail": true,
    "email": false,
    "line_notify": true
  }
]
```

## 比較函數 compare\_function

當 event\_type="device" 時, 設定觸發的條件, 語言為 javascript, 內建有兩個參數

- device:Device - 觸發的裝置
- data:any - 裝置傳來的資料

下面是範例

```
device.device_name=='device1' && data.temperature>50
```

## 訊息文字 message

下面是範例

```
裝置${device.device_name}的溫度過高, 目前是${data.temperature}度
```

# GraphQL

---

## 維修單查詢範例

### Request

```
query {
  domains {
    domain_name
    users {
      user_name
    }
    devices{
      device_name
    }
    repair_orders{
      custom_description
      repair_description
      repair_items {
        description
      }
      repair_records {
        description
        repair_item {
          description
        }
        repair_user_records{
          user{
            user_name
          }
        }
      }
    }
  }
}
```

### Response

```
{
  "data": {
    "domains": [
      {
        "domain_name": "root",
        "users": [
          {
            "user_name": "root 管理員"
          }
        ],
        "devices": [],
        "repair_orders": []
      },
      {
        "domain_name": "台北市",
        "users": [
```

```
{
  {
    "user_name": "台北管理員"
  }
],
"devices": [],
"repair_orders": []
},
{
  "domain_name": "台中市",
  "users": [],
  "devices": [],
  "repair_orders": []
},
{
  "domain_name": "台南市",
  "users": [
    {
      "user_name": "台南管理員"
    },
    {
      "user_name": "工程師1"
    },
    {
      "user_name": "訂單管理員1"
    }
  ],
  "devices": [
    {
      "device_name": "AA-70-2209-01-0079-377"
    },
    {
      "device_name": "AA-70-2211-01-0155-160"
    },
    {
      "device_name": "AA-70-2211-01-0217-874"
    },
    {
      "device_name": "AA-70-2211-01-0225-889"
    },
    {
      "device_name": "AA-70-2212-01-0028-070"
    },
    {
      "device_name": "AA-70-2212-01-0092-973"
    }
  ],
  "repair_orders": [
    {
      "custom_description": "發電機無法充電",
      "repair_description": "",
      "repair_items": [],
      "repair_records": []
    }
  ]
},
{
  "domain_name": "高雄市",
  "users": [],
  "devices": [],
```

```
    "repair_orders": []
  },
  {
    "domain_name": "新北市",
    "users": [],
    "devices": [],
    "repair_orders": []
  },
  {
    "domain_name": "桃園市",
    "users": [],
    "devices": [],
    "repair_orders": []
  },
  {
    "domain_name": "萬華區",
    "users": [],
    "devices": [],
    "repair_orders": []
  },
  {
    "domain_name": "內湖區",
    "users": [],
    "devices": [],
    "repair_orders": []
  },
  {
    "domain_name": "永康區",
    "users": [
      {
        "user_name": "台南永康管理員"
      },
      {
        "user_name": "維修人員1"
      },
      {
        "user_name": "維修人員2"
      }
    ],
    "devices": [],
    "repair_orders": [
      {
        "custom_description": "讀取數值時好時壞",
        "repair_description": "訊號線過長",
        "repair_items": [],
        "repair_records": []
      }
    ]
  },
  {
    "domain_name": "安南區",
    "users": [
      {
        "user_name": "維修人員3"
      }
    ],
    "devices": [],
    "repair_orders": [
      {
```

```

    "custom_description": "太陽能無法充電",
    "repair_description": "太陽能接線鬆脫",
    "repair_items": [],
    "repair_records": []
  },
  {
    "custom_description": "機台無法開機",
    "repair_description": "主控版故障",
    "repair_items": [
      {
        "description": "前往廠區檢查"
      },
      {
        "description": "更換主機板"
      }
    ],
    "repair_records": [
      {
        "description": "前去檢查發現主機板出問題",
        "repair_item": {
          "description": "前往廠區檢查"
        },
        "repair_user_records": [
          {
            "user": {
              "user_name": "維修人員1"
            }
          }
        ]
      },
      {
        "description": "把壞掉的主機板換掉",
        "repair_item": {
          "description": "更換主機板"
        },
        "repair_user_records": [
          {
            "user": {
              "user_name": "維修人員1"
            }
          },
          {
            "user": {
              "user_name": "維修人員2"
            }
          }
        ]
      }
    ]
  }
]
}

```

Domain 內資料查詢範例

## Request

```
query{
  domain(domain_id:118) {
    domain_name
    child_domains{
      domain_name
    }
    all_domains{
      domain_name
    }
    all_users{
      user_name
    }
    all_repair_orders {
      repair_description
    }
    all_devices {
      device_name
    }
  }
}
```

## Response

```
{
  "data": {
    "domain": {
      "domain_name": "X公司",
      "child_domains": [
        {
          "domain_name": "台北"
        }
      ],
      "all_domains": [
        {
          "domain_name": "X公司"
        },
        {
          "domain_name": "台北"
        },
        {
          "domain_name": "內湖"
        }
      ],
      "all_users": [
        {
          "user_name": "管理員"
        },
        {
          "user_name": "訪客"
        }
      ],
      "all_repair_orders": [
        {
          "repair_description": "太陽能板接頭鬆脫"
        }
      ]
    }
  }
}
```



```

    }
  ],
  "all_devices": [
    {
      "device_name": "AA-70-2212-01-0076-123"
    },
    {
      "device_name": "AA-70-2209-01-0079-567"
    }
  ]
}
}
}

```

## Device 連線狀態查詢

### Request

```

query{
  device_types{
    device_type_id
    device_type_name
  }
  domain(domain_id:118) {
    pomcubes:all_devices(device_type_id:6) {
      device_name
      is_online
    }
    pv_inverters:all_devices(device_type_id:8) {
      device_name
      is_online
    }
  }
}

```

### Response

```

{
  "data": {
    "device_types": [
      {
        "device_type_id": 6,
        "device_type_name": "pomcube"
      },
      {
        "device_type_id": 8,
        "device_type_name": "pv_inverter"
      }
    ],
    "domain": {
      "pomcubes": [
        {
          "device_name": "AA-70-2212-01-0076-789",
          "is_online": true
        },

```

```
{
  "device_name": "AA-70-2209-01-0079-321",
  "is_online": false
},
],
"pv_inverters": [
  {
    "device_name": "EMS000002",
    "is_online": true
  },
  {
    "device_name": "EMS000003",
    "is_online": true
  }
]
}
}
```

# 裝置資料查詢與寫入 (Data API)

## 簡介

透過 Data API，使用者可以取得裝置的相關數據，並利用這些資料提供全面的能源管理與分析能力，它包括以下內容：

1. 即時/歷史太陽能功率
2. 即時/歷史市電功率
3. 即時/歷史負載功率
4. 即時/歷史電池電量 (SOC)
5. 根據時間查詢該站點的太陽能發電預測

## Data API 路徑

GET /api/device\_datas/query

## 使用流程

1. 使用 POST /auth/login 登入
  1. 帳號 (user\_account): "admin"
  2. 密碼 (user\_password): "xxx"
2. 利用 GET /api/device\_datas/query 查詢設備資料

## Data API - GET "/api/device\_datas/query"

- API說明：
  - 查詢裝置資料
- HTTP Header：
  - Authorization: login 傳回來的 access\_token。例如: "Bearer ey.123.xx"
- 參數說明
  - organization\_id: (必填) 組織 ID
  - device\_type\_name: 裝置類型名稱
  - device\_output\_name: (必填) 輸出埠名稱。目前必須是下面兩種
    - "pomcube\_data": pomcube的資料
    - "solar\_prediction": 太陽能預測的資料
  - domain\_id: 查詢此 domain 所有 device, 忽略時等同於 organization\_id
  - device\_names: 要查詢的 device 名稱, 多個裝置以 "," 隔開, ex: "device1,device2,device3", 忽略時等同於 organization\_id 下所有裝置
  - place\_names: 要查詢的 place 名稱, 多個 place 以 "," 隔開, ex: "place1,place2,place3"
  - start: 開始時間, ex: "2023-09-01T00:00:00.000Z", "1621726200", "-10s", "-30m", "-6h", "-7d", "-1mo", "-1y"
  - stop: 結束時間, ex: "2023-09-01T00:00:00.000Z"
  - fields: (必填) 要查詢的欄位, 每個欄位用 "," 隔開. ex: "field1,field2"
  - where: 查詢條件. 例如 '{"domain\_id":"1","device\_name":["device1","device2"]}' 代表 (domain\_id==1 && (device\_name=="device1" || device\_name=="device2"))。查詢欄位必須是 tag
  - differenceNonNegativeSource: 是否在統計前, 先把後一筆的值減去前一筆的值, 若減完結果為負值則為 null。可為 "false" 或 "true"。預設 "false"

- group\_by: 需要群組的欄位，欄位名稱用 ',' 隔開。若欄位名稱開頭有 '!'，則此欄位僅在統計前群組，統計後不群組。ex: "!DeviceName,\_field"
- every: 每段統計的時間區間, 需搭配 "time\_function" 一起使用
- create\_empty: 每段統計的時間區間值為 null 時, 是否仍建立此時間。可為 "false" 或 "true"。預設 "false"
- time\_src: 每段統計的時間要使用 開頭時間('\_start') 或 結束時間('\_stop')。ex: one of ["\_start", "\_stop"]
- timezone: 當使用 every 或其他具時區的查詢時用到，例如 "每日"
- difference: 是否將後一時間段的值減去前一時間段的值。可為 "false" 或 "true"。預設 "false"
- time\_function: "每段時間" 的統計 function。ex: one of ['first', 'last', 'min', 'max', 'mean', 'median', 'count', 'sum', 'spread','integral']
  - last: 最後一筆
  - min: 最小
  - mean: 平均值
  - count: 個數
  - sum: 總計
  - spread: 區間內 (最大-最小)
  - integral: 積分 (值\*秒差)
- group\_function: 將 "多個群組" 合併成 "一個時間群組"。ex: one of ['min', 'max', 'mean', 'median', 'count', 'sum']
- pivot\_columns: 是否將不同 table 的值，依據時間區間，合併 欄位(ex:"\_field") 或 tag(ex:"LoadPower") 成為一筆紀錄。ex: "\_field"
- limit: 每個 table 最多幾筆
- output\_format: 輸出格式。one of ['json', 'influx\_line\_protocol']. default is 'json'
- output\_zip: 是否輸出壓縮檔。one of ['true', 'false']. default is 'false'
- debug: 偵錯使用。可為 "false" 或 "true"。預設 "false"

## 傳回範例

- 一般傳回格式

```
[
  {
    "table": 0,
    "_time": "2024-07-03T08:57:01.303Z",
    "_value": 673,
    "DeviceName": "AA-70-2211-01-0217-874",
    "_field": "LoadPower"
  },
  {
    "table": 1,
    "_time": "2024-07-03T08:57:01.303Z",
    "_value": 863,
    "DeviceName": "AA-70-2211-01-0217-874",
    "_field": "SolarPower"
  }
]
```

- pivot\_columns = "\_field"

```
[
  {
    "table": 0,
    "_time": "2024-07-03T08:51:01.007Z",
```

```

    "DeviceName": "AA-70-2211-01-0217-874",
    "LoadPower": 655,
    "SolarPower": 539
  }
]

```

## Pomcube 相關欄位 (device\_output\_name="pomcube\_data")

- **GenPower**: 發電機功率 (W)，顯示當前發電機的輸出功率。
- **GridPower**: 市電功率 (W)，表示從市電網獲得的功率。
- **LoadPower**: 負載功率 (W)，表示當前負載消耗的功率。
- **SolarPower**: 太陽能功率 (W)，顯示太陽能板產生的功率。
- **GenEnergy**: 發電機累積度數 (kWh)，表示發電機自安裝以來產生的總電量。
- **GridEnergy**: 市電累積度數 (kWh)，顯示從市電網累積獲得的總電量。
- **LoadEnergy**: 負載累積度數 (kWh)，代表負載自安裝以來消耗的總電量。
- **SolarEnergy**: 太陽能累積度數 (kWh)，表示太陽能板自安裝以來產生的總電量。
- **BatSoC**: 電池容量 (%)，顯示當前電池的剩餘容量比例。
- **BatCurr**: 總電流 (A)，表示系統的總電流。
- **BatVolt**: 總電壓 (V)，表示系統的總電壓。
- **BatTemp**: 電池溫度 (degC)，表示電池區的溫度。
- **EnvTemp1 & EnvTemp2**: 環境溫度 1 & 2 (degC)，分別表示兩個不同位置的環境溫度。
- **EnvHumidity1 & EnvHumidity2**: 環境濕度 1 & 2 (%), 分別表示兩個不同位置的環境濕度。
- **EnvCO & EnvCO2**: 一氧化碳和二氧化碳濃度 (PPM)，分別表示當前空氣中的 CO 和 CO2 濃度。
- **Irradiance**: 日照功率，太陽光照射到太陽能板的強度，通常表示為瓦特每平方米 (W/m<sup>2</sup>)
- **PR**: Performance Ratio 效能比，意即用來評估一個太陽能電站的效能表現。

## 太陽能預測相關欄位 (device\_output\_name="solar\_prediction")

- **DirectRadiation**: 日照功率，太陽光照射到太陽能板的強度，通常表示為瓦特每平方米 (W/m<sup>2</sup>)
- **SolarArea**: 太陽能板面積 (m<sup>2</sup>)
- **SolarEff**: 太陽能板發電效率 (0~1) (0%~100%)
- **SolarPower**: 發電功率 (W)

## 使用範例

### 太陽能預測 (站點)

```

{
  organization_id: 2,
  domain_id: 4,
  device_output_name: "solar_prediction",
  start: '2024-06-26T16:00:00Z',
  stop: '2024-06-28T16:00:00Z',
  fields: "SolarPower",
  timezone: "Asia/Taipei",
  group_by: "_time",
  group_function: "sum"
}

```

### 太陽能預測 (單台)

```
{
  organization_id: 2,
  device_names: 'AA-70-2212-01-0028-070',
  device_output_name: "solar_prediction",
  start: '2024-06-26T16:00:00Z',
  stop: '2024-06-28T16:00:00Z',
  fields: "DirectRadiation,SolarArea,SolarEff,SolarPower",
  timezone: "Asia/Taipei",
  pivot_columns: "_field" // 此欄位非必須
}
```

每日太陽能度數預測 (單台日照量轉換 "W/m<sup>2</sup>" -> "Ws/m<sup>2</sup>")

- 將輸出結果 DirectRadiation / (60\*60) 就可以把 "Ws/m<sup>2</sup>" 轉成 "Wh/m<sup>2</sup>"

```
{
  organization_id: 2,
  device_names: 'AA-70-2212-01-0028-070',
  device_output_name: "solar_prediction",
  start: '2024-06-26T16:00:00Z',
  stop: '2024-06-28T16:00:00Z',
  fields: "DirectRadiation",
  every: "1d",
  timezone: "Asia/Taipei",
  time_function: "integral",
  time_src: '_start',
  pivot_columns: "_field" // 此欄位非必須
}
```

目前功率 (單台)

```
{
  organization_id: 2,
  device_names: 'AA-70-2212-01-0028-070',
  device_output_name: "pomcube_data",
  start: 0,
  fields: "SolarPower,LoadPower",
  time_function: "last",
  group_by: "_field"
}
```

目前太陽能發電功率 (站點)

```
{
  organization_id: 2,
  domain_id: 4,
  device_output_name: "pomcube_data",
  start: 0,
  fields: "SolarPower",
  time_function: "last",
  group_by: "!DeviceName,_field",
}
```

```
group_function: "sum"
}
```

### 每日歷史功率圖 (單台)

```
{
  organization_id: 2,
  device_names: 'AA-70-2212-01-0028-070',
  device_output_name: "pomcube_data",
  start: '2024-05-31T16:00:00Z',
  stop: '2024-06-30T16:00:00Z',
  fields: "SolarPower,LoadPower,GenPower,GridPower",
  every: "1d",
  timezone: "Asia/Taipei",
  time_function: "mean",
  create_empty: "true", // 此欄位非必須
  time_src: '_start',
  group_by: "_field",
  pivot_columns: "_field" // 此欄位非必須
}
```

### 每日歷史功率圖 (站點)

```
{
  organization_id: 2,
  domain_id: 4,
  device_output_name: "pomcube_data",
  start: '2024-05-31T16:00:00Z',
  stop: '2024-06-30T16:00:00Z',
  fields: "SolarPower,LoadPower,GenPower,GridPower",
  every: "1d",
  timezone: "Asia/Taipei",
  time_function: "mean",
  create_empty: "true", // 此欄位非必須
  time_src: '_start',
  group_by: "!DeviceName,_field",
  group_function: "sum",
  pivot_columns: "_field" // 此欄位非必須
}
```

### 每日電表累計度數 (單台)

```
{
  organization_id: 2,
  device_names: 'AA-70-2212-01-0028-070',
  device_output_name: "pomcube_data",
  start: '2024-05-30T16:00:00Z',
  stop: '2024-06-30T16:00:00Z',
  fields: "SolarEnergy,LoadEnergy",
  every: "1d",
  timezone: "Asia/Taipei",
  time_function: "last",
}
```

```
create_empty: "true", // 此欄位非必須
group_by: "_field",
pivot_columns: "_field" // 此欄位非必須
}
```

#### 每日太陽能發電度數及用電度數 (單台)

```
{
  organization_id: 2,
  device_names: 'AA-70-2212-01-0028-070',
  device_output_name: "pomcube_data",
  start: '2024-05-30T16:00:00Z',
  stop: '2024-06-30T16:00:00Z',
  fields: "SolarEnergy,LoadEnergy",
  differenceNonNegativeSource: "true",
  every: "1d",
  timezone: "Asia/Taipei",
  time_function: "sum",
  create_empty: "true",
  time_src: '_start',
  group_by: "_field",
  pivot_columns: "_field" // 此欄位非必須
}
```

#### 每月太陽能發電度數及用電度數 (站點)

```
{
  organization_id: 2,
  domain_id: 4,
  device_output_name: "pomcube_data",
  start: '2023-11-30T16:00:00Z',
  stop: '2024-12-31T16:00:00Z',
  fields: "SolarEnergy,LoadEnergy",
  differenceNonNegativeSource: "true",
  every: "1mo",
  timezone: "Asia/Taipei",
  time_function: "sum",
  create_empty: "true",
  time_src: '_start',
  group_by: "!DeviceName,_field",
  group_function: "sum",
  pivot_columns: "_field" // 此欄位非必須
}
```

#### 每日太陽能發電度數及用電度數 (單台)

```
{
  organization_id: 2,
  device_names: 'AA-70-2212-01-0028-070',
  device_output_name: "pomcube_data",
  start: '2024-05-30T16:00:00Z',
  stop: '2024-06-30T16:00:00Z',
```



```

fields: "SolarEnergy,LoadEnergy",
differenceNonNegativeSource: "true",
every: "1d",
timezone: "Asia/Taipei",
time_function: "sum",
create_empty: "true",
time_src: '_start',
group_by: "_field",
pivot_columns: "_field" // 此欄位非必須
}

```

最近24小時太陽能發電度數及用電度數 (站點)

```

{
  organization_id: 2,
  domain_id: 4,
  device_output_name: "pomcube_data",
  start: '-24h',
  fields: "SolarEnergy,LoadEnergy",
  differenceNonNegativeSource: "true",
  timezone: "Asia/Taipei",
  time_function: "sum",
  group_by: "!DeviceName,_field",
  group_function: "sum",
}

```

## 其他 API

GET /api/device\_datas/curtailment\_ratio

計算棄電比，間隔(every) 最小為 30m

下面是8月1日到8月3日的棄電比範例:

參數：

```

{
  organization_id: 118,
  domain_id: 123,
  device_output_name: 'pomcube_data',
  start: '2024-07-31T16:00:00Z',
  stop: '2024-08-03T16:00:00Z',
  every: '1d',
  time_src: '_start',
  timezone: 'Asia/Taipei'
}

```

傳回

```

{
  "times": [
    "2024-07-31T16:00:00Z",
    "2024-08-01T16:00:00Z",

```

```

    "2024-08-02T16:00:00Z"
  ],
  "solar_energys": [
    12.400999999999975,
    13.8610000000000118,
    13.5410000000000036
  ],
  "prediction_energys": [
    98.652329999999996,
    103.138604999999996,
    102.132764999999996
  ],
  "curtailment_ratios": [
    87.42959238773177,
    86.5608032995985,
    86.74176695402299
  ]
}

```

GET /api/device\_datas/pr

計算PR值，間隔(every) 最小為 1h

下面是8月1日到8月3日的 PR 值範例:

參數：

```

{
  organization_id: 118,
  domain_id: 126,
  device_output_name: 'hermes_report',
  start: '2024-07-31T16:00:00Z',
  stop: '2024-08-03T16:00:00Z',
  every: '1d',
  time_src: '_start',
  timezone: 'Asia/Taipei'
}

```

傳回

```

[
  {
    "_time": "2024-07-31T16:00:00Z",
    "SolarEnergy": 6526.429,
    "SolarCapacity": 1500,
    "Irradiance": 5147.997,
    "PR": 84.52
  },
  {
    "_time": "2024-08-01T16:00:00Z",
    "SolarEnergy": 7574.601,
    "SolarCapacity": 1500,
    "Irradiance": 5323.197,
    "PR": 94.86
  },

```

```
{
  "_time": "2024-08-02T16:00:00Z",
  "SolarEnergy": 7201.811,
  "SolarCapacity": 1500,
  "Irradiance": 4938.476,
  "PR": 97.22
}
```

GET /api/device\_datas/device\_event

取得裝置事件

下面是取得裝置事件的範例:

參數：

```
{
  organization_id: 118,
  device_names: 'EMS000003',
  start: '2024-08-05T10:00:00Z',
  stop: '2024-08-05T10:05:00Z',
  time_src: '_start',
  timezone: 'Asia/Taipei'
}
```

傳回

```
[
  {
    "_time": "2024-08-05T10:00:33.387Z",
    "DeviceName": "EMS000003",
    "DeviceType": "pv_inverter",
    "ErrorCode": "E1004",
    "ErrorDescription": "E-dRag排程狀態發生錯誤"
  },
  {
    "_time": "2024-08-05T10:04:33.368Z",
    "DeviceName": "EMS000003",
    "DeviceType": "pv_inverter",
    "ErrorCode": "E1001",
    "ErrorDescription": "Aggregator聚合器運行發生錯誤"
  }
]
```

GET /api/device\_datas/report

說明

產生報表

參數說明

Name	Description
------	-------------

Name	Description
report_type	報表類型: "pv" 或 "ess"
time_type	年報或月報 "year" 或 "month"
domain_id	域ID
device_type_id	裝置類型
year	年 ex: 2024
month	月 0:一月, 11:十二月
timezone	時區 ex: "Asia/Taipei"

GET /api/feed\_in\_tariff

說明

取得此 domain 的躉售費率表

參數說明

Name	Description
domain_id	域ID

傳回範例

```
[
  {
    "time": "2023-12-31T16:00:00.000Z",
    "tariff": 4.5
  },
  {
    "time": "2024-05-31T16:00:00.000Z",
    "tariff": 5
  },
  {
    "time": "2024-12-31T16:00:00.000Z",
    "tariff": null
  }
]
```

上面的傳回結果代表 2024 年  
1月1日~5月31日的躉售費率是每度4.5元。  
6月1日~12月31日的躉售費率是每度5元。

GET /api/generate\_money

說明

取得此 domain 賺了多少錢

參數說明

Name	Description
domain_id	域ID
device_type_id	裝置類型
start_time	開始時間
end_time	結束時間

#### 傳回範例

```
{
  "tariffs": [
    {
      "time": "2023-12-31T16:00:00.000Z",
      "tariff": 5
    },
    {
      "time": "2024-12-31T16:00:00.000Z",
      "tariff": null
    }
  ],
  "incomes": [
    {
      "time": "2024-06-30T16:00:00.000Z",
      "tariff": 5,
      "diff_energy": 2500,
      "amount": 12500
    },
    {
      "time": "2024-07-31T16:00:00.000Z",
      "tariff": 5
    }
  ],
  "total_income": 12500
}
```

上面的傳回結果代表7月總共產生了2500度電, 每度5元, 共賺了12500元

## 裝置資料的寫入

- API: POST /api/device\_datas/{organization\_id}/{measurement\_name}/write
- 說明: 用來寫入裝置資料
- 參數:
  - organization\_id: 組織 ID
  - measurement\_name: 資料表名稱
  - [post data]: 一層結構的 JSON 物件陣列, 例如: "[{'DeviceName':'device1','temp':26.3,'humi':35.2}, {'DeviceName':'device2','temp':27.9,'humi':33.6}]" , DeviceName 必須為存在的裝置