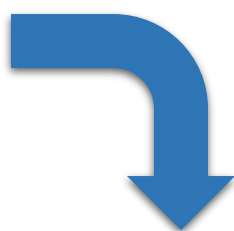
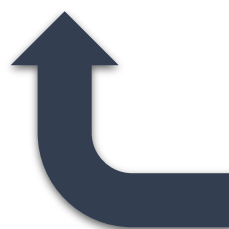


openHAB 應用在 IP POWER (Windows 專用)

手冊更新:2024/06 Ver.1.1



openHAB
empowering the smart home



目錄

| | |
|--|----|
| 1. openHAB 安裝準備 | 4 |
| 1.1 安裝附加元件..... | 10 |
| 1.2 IP Power 配置前準備 | 12 |
| 2. 配置 IP Power 適用 9255, 9258 系列..... | 13 |
| 2.1 建立 things..... | 13 |
| 2.2 建立 Channel | 17 |
| 2.3 創建 Item | 21 |
| 2.4 9258 系列配置參數對應的 code..... | 25 |
| 3. 配置 IP Power 適用 9850, 9858, 9855 系列..... | 26 |
| 3.1 創建 things..... | 26 |
| 3.2 創建 Channels | 28 |
| 3.3 創建 item..... | 30 |
| 3.4 98 系列配置參數對應的 code..... | 30 |
| 3.5 創建屬性 Channel..... | 31 |
| 4. MQTT 設定 | 36 |
| 4.1 創建 MQTT Broker..... | 37 |
| 4.2 創建 Generic MQTT Thing | 38 |
| 4.3 創建 Generic MQTT Thing 的 Channels | 39 |
| 4.4 建立 items..... | 42 |
| 4.5 MQTT 的 Code | 42 |
| 5. openHAB App 內網(LAN)和外網(WAN)設定: | 43 |
| 6. 展示手冊圖片中 IP Power 的 code..... | 51 |
| 6.1 9258 code..... | 51 |
| 6.2 98 系列 code | 52 |
| 6.3 Generic MQTT thing code | 54 |
| 6.4 屬性設定: (如: 電流, 電壓, 溫度) | 56 |

AVIOSYS 的 IP POWER 現在可應用在 openHAB，在此系統中您可依自身的需求去設定您的 IP POWER。請依照內文依序操作：

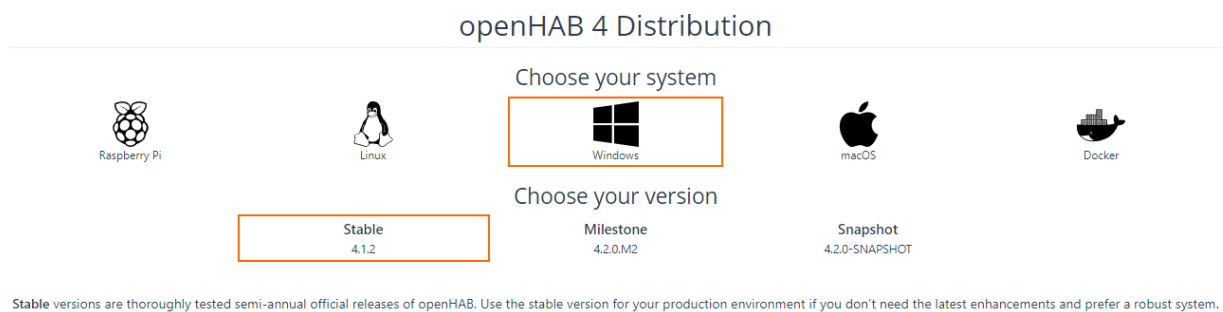
1. openHAB 安裝準備

請移至 openHAB 官方下載頁面，點選 Windows 版本。並依照指示開始安裝。

官方連結: <https://www.openhab.org/download/>

當您點選要安裝的 OS 時，下方會出現三個版本的 openHAB。

本次教學以 Stable Ver.4.1.2 為主



請照官方指示先下載 JAVA 17 平台，並下載 Azul Zulu Builds of OpenJDK。

可點擊官方提供的連結進行下載



Manual Installation

1. Install a recent Java 17 platform (we recommend the Zulu builds of OpenJDK)

***注意：**此安裝為必要條件，若不安裝此平台將無法啟用 openHAB。

下載並安裝完成後，請依照下一頁的指示完成 Azul 設定。

Azul 設定：

點擊桌面上的 Windows ，並點選設定圖示 ，再點選系統。

進入系統的視窗後，點擊左邊的關於(1)，進入關於的頁面，

再點擊“進階系統設定(2)”。

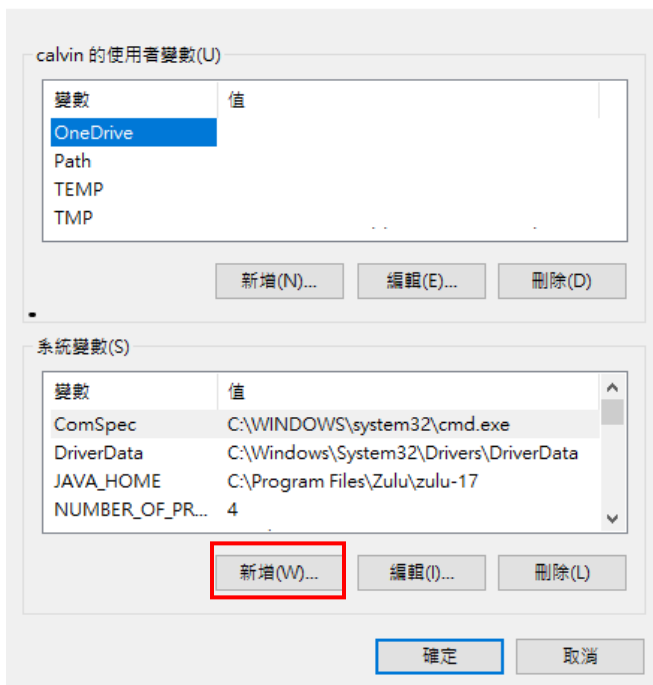


進入 進階系統設定 後，請點擊“環境變數”。



進入環境變數後，請點擊下方的“新增”，點擊後會出現“新增系統變數”。

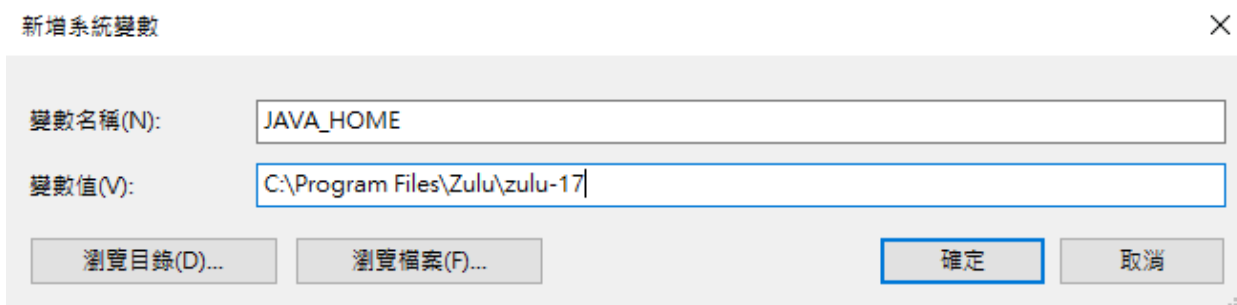
環境變數



出現新增系統變數後，請在 變數名稱 和 變數值 填入以下資訊。

- (1) 變數名稱: JAVA_HOME
- (2) 變數值: 您安裝 Azul Zulu 的磁碟位置。

輸入完畢後，點擊下方 OK 後即可完成 Azul Zulu 的安裝。



openHAB 安裝:

回到 openHAB 的安裝頁面，開始進行下一個步驟。

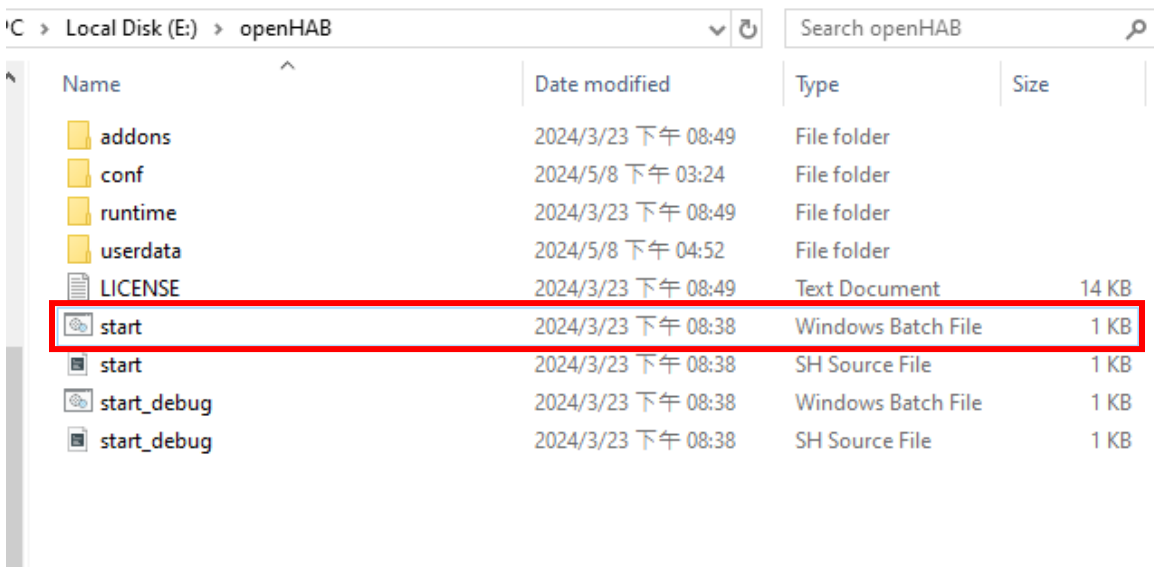
- (1) 接著我們再點擊下載專區的“Download openHAB 4.1.2 Stable Runtime”。
下載完成後開始進行安裝。

安裝完成後，請先建立 openHAB 的資料夾，並選擇您要放入的磁碟槽。

進入 openHAB 的資料夾後, 請解壓縮 openHAB 的 zip 下載檔案.

解壓縮完成後會如下圖所示, 確認解壓縮完成後,

點擊資料夾裡的 start(Windows Batch File).

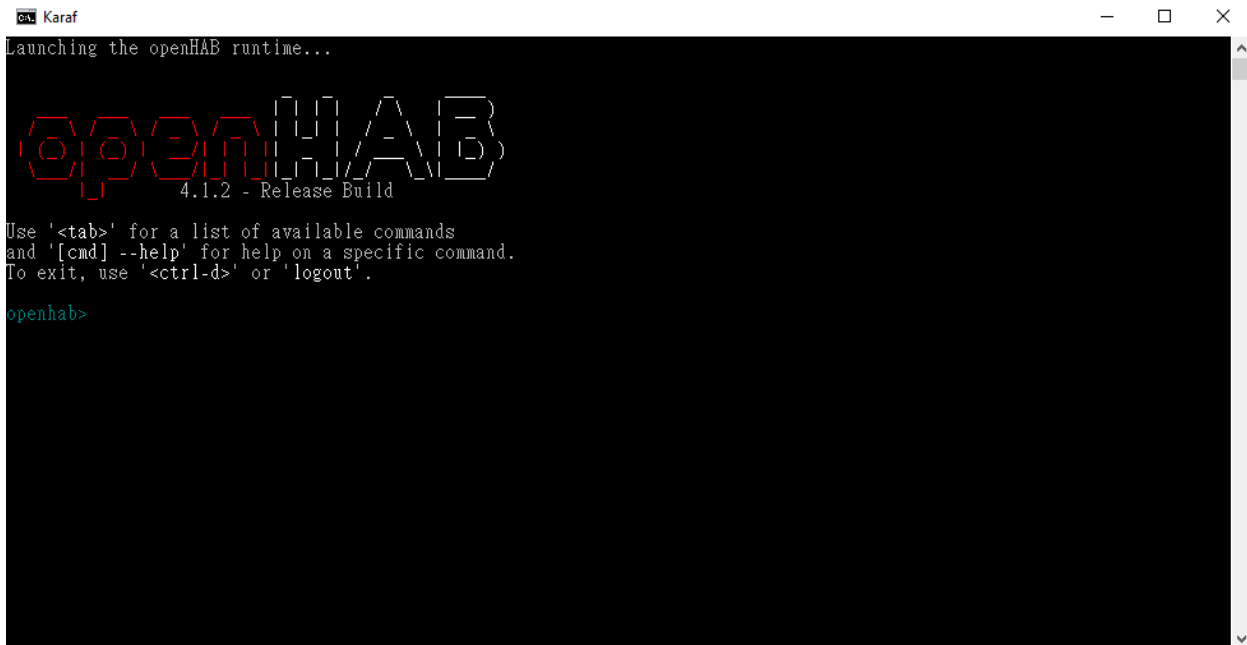


程式開啟時, 需要等待大約 10~30 秒與 openHAB 連線.

***注意:** 若長時間都維持此畫面, 代表程式沒有正常運行. 請檢查前面步驟或重啟 start.



運作成功後會顯示 openHAB 的圖示, 請在使用 openHAB 期間不要關閉此程式.




```
Karaf
Launching the openHAB runtime...

  openHAB
  4.1.2 - Release Build

Use '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
To exit, use '<ctrl-d>' or 'logout'.

openh>
```

請開啟任何一個瀏覽器(如:Edge, Chrome), 並輸入: <http://localhost:8080>
進入網頁後, 先建立您的帳戶 openHAB 帳戶.



Create a first administrator account to continue.

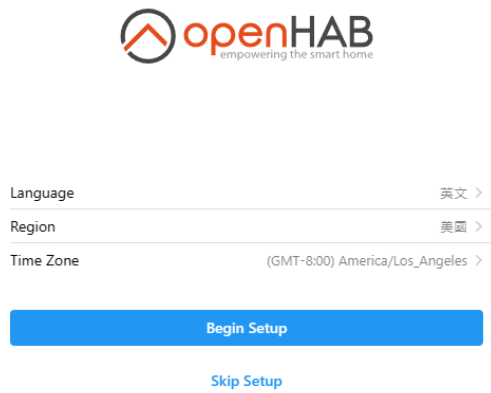
User Name

Password

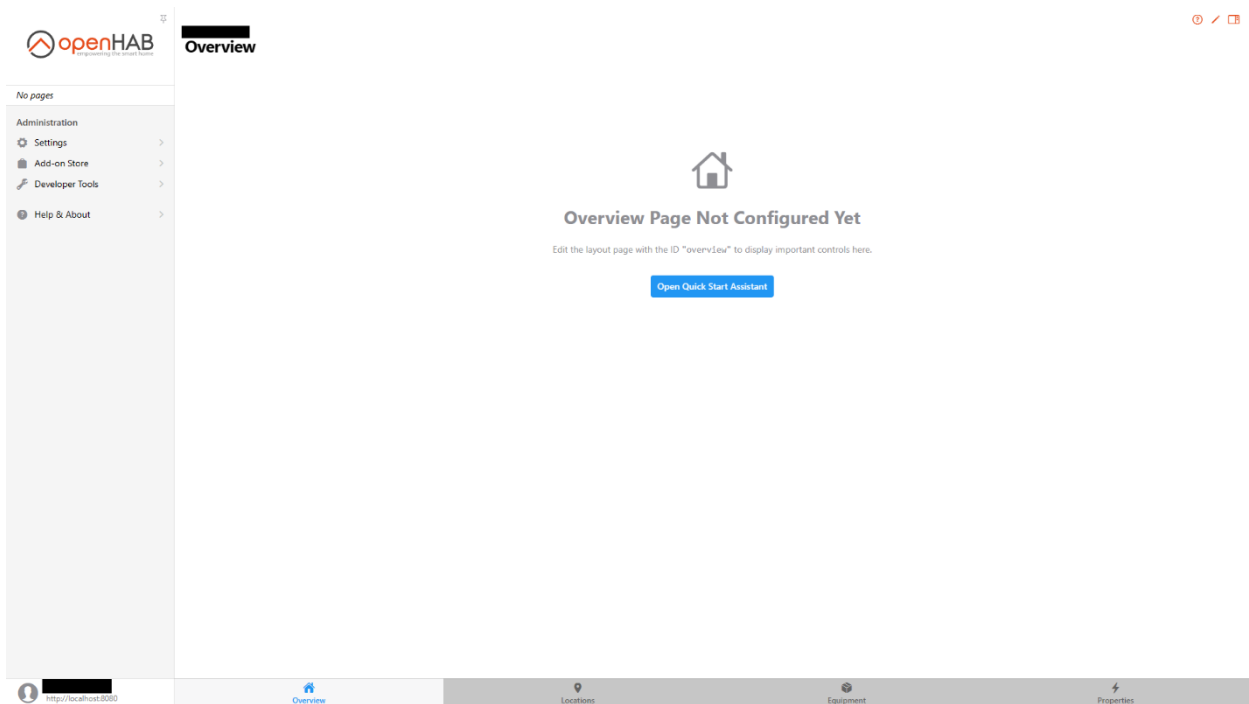
Confirm New Password

[Create Account](#)

建立完成後會進入配置頁面，您不需要立刻做前置設定，可先跳過前置設定直接進入主頁面。

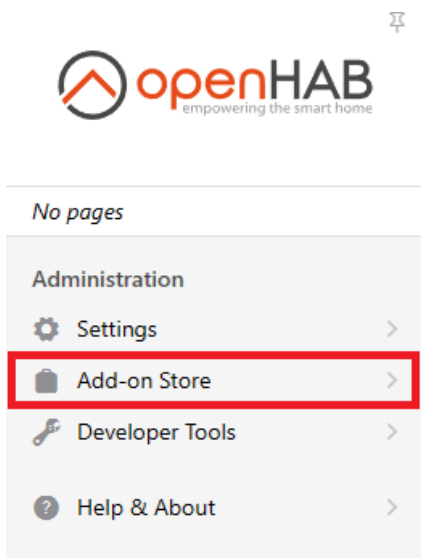


當您進入到此畫面時，代表您已成功建立您的 openHAB。

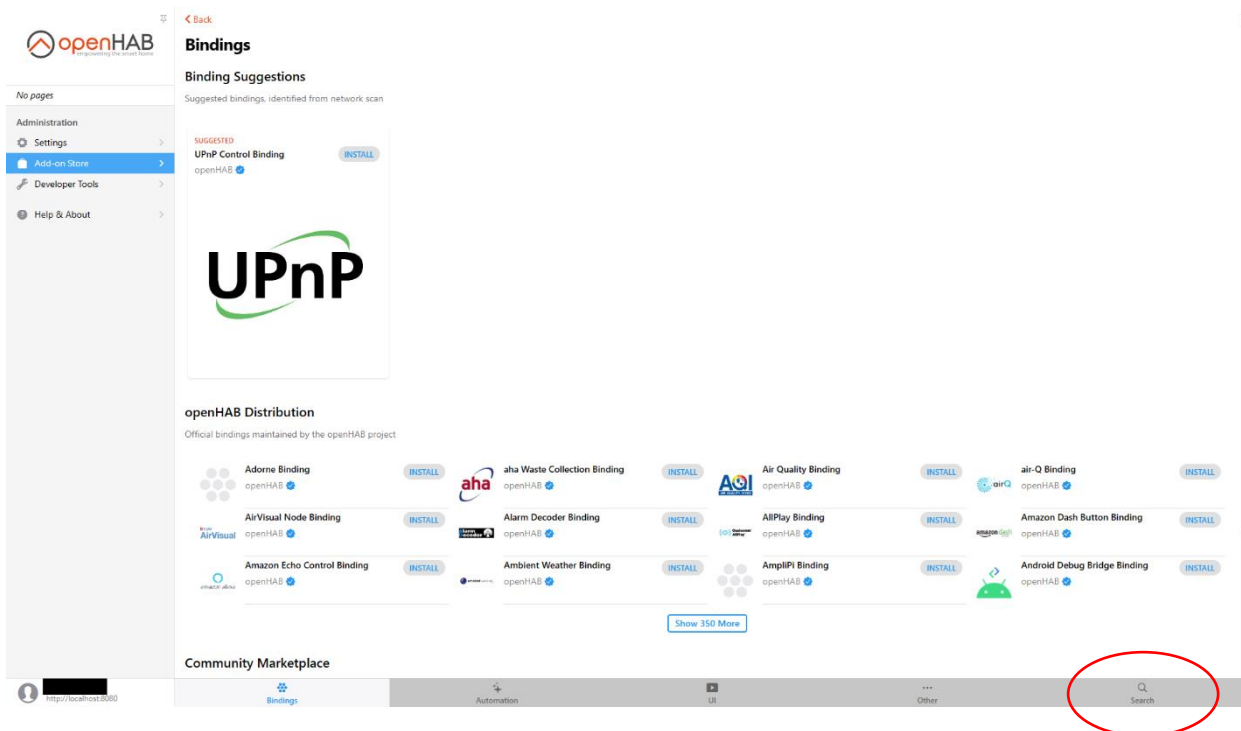


1.1 安裝附加元件

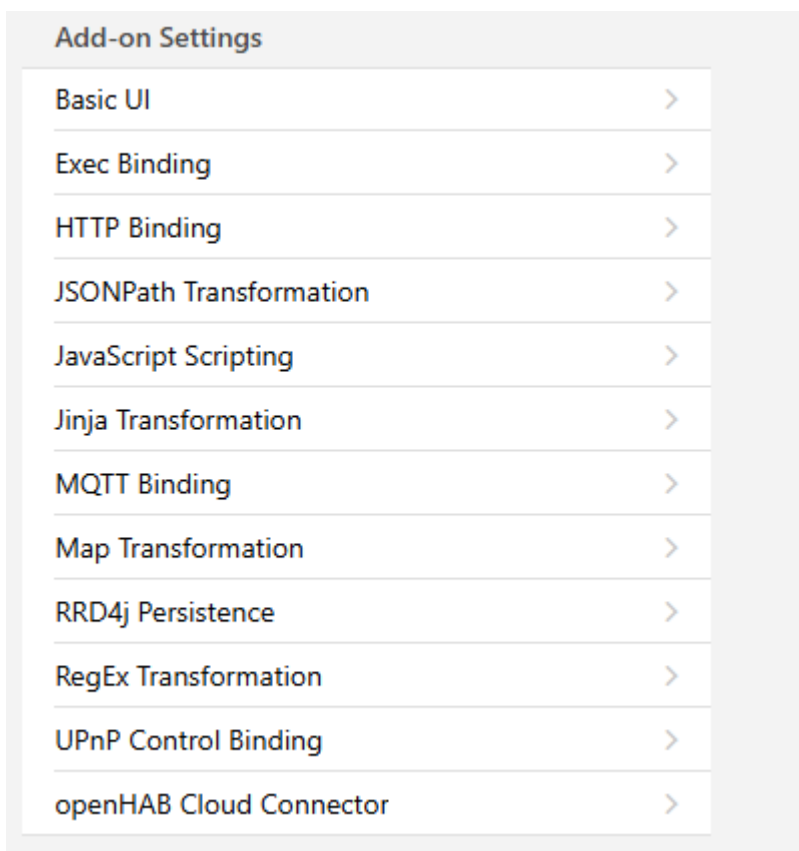
在開始設定 IP Power 至 openHAB 之前, 需要事先安裝一些附加元件(Bindings)
請先點擊主頁面左邊的 “Add-on Store”



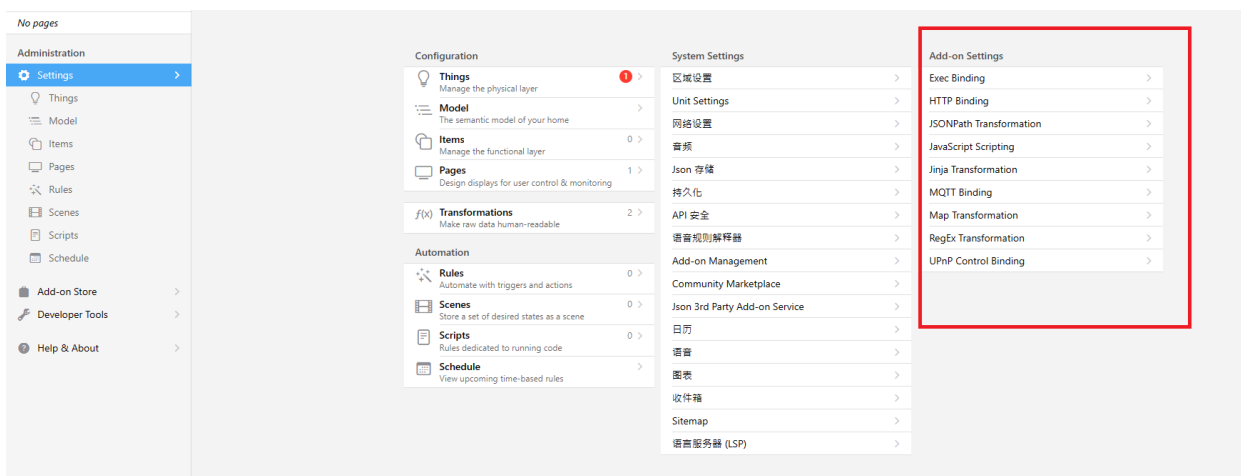
進入頁面之後, 請點擊右下角的 Search.



進入 Search 頁面後，請依照下方列表所列出的元件進行安裝。

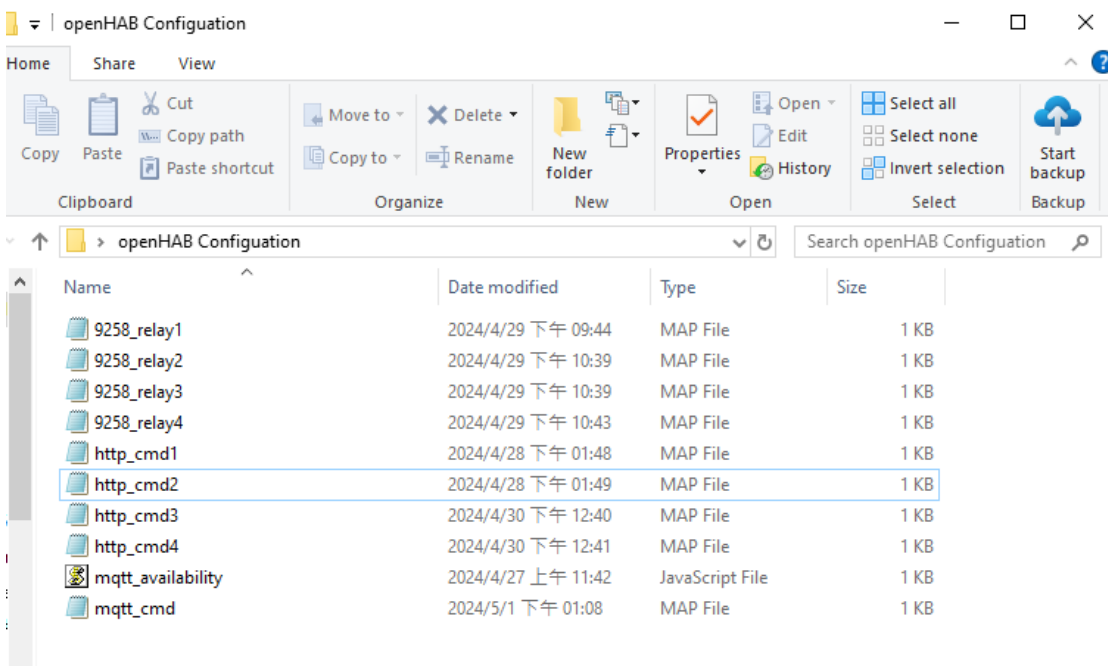


安裝完畢後，所有的附加元件都會顯示在 Settings 介面裡的 “Add on Settings” (紅框)

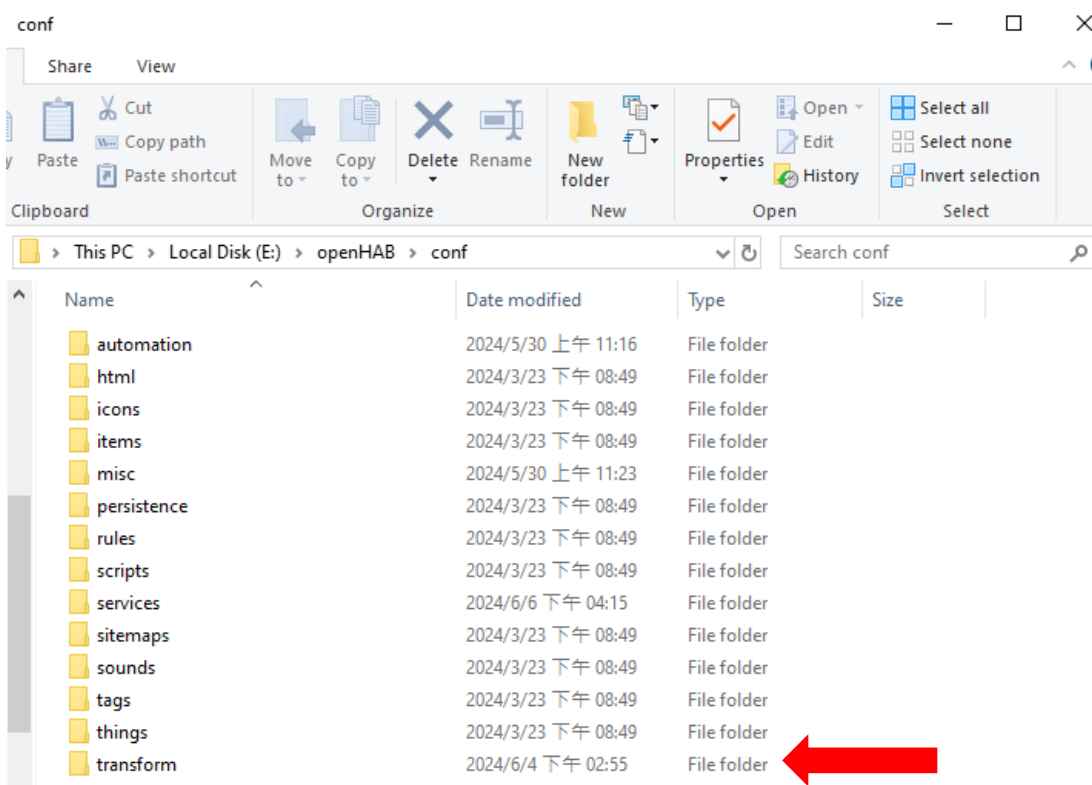


1.2 IP Power 配置前準備

請將在官網下載手冊的檔案中，點選 **openHAB configuration**，並將這些檔案先放置 **openHAB/conf/transform**，若無放入將無法完成後續的裝置設定。



openHAB/conf/transform 位置

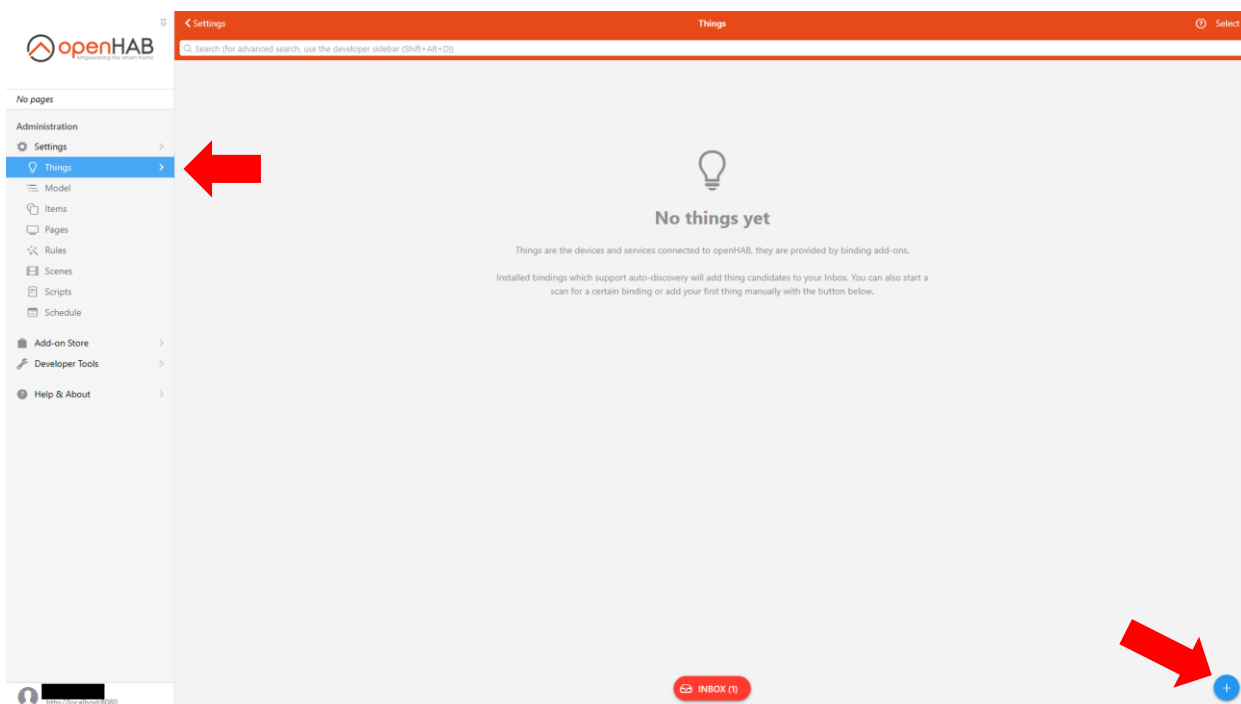


2. 配置 IP Power 適用 9255, 9258 系列

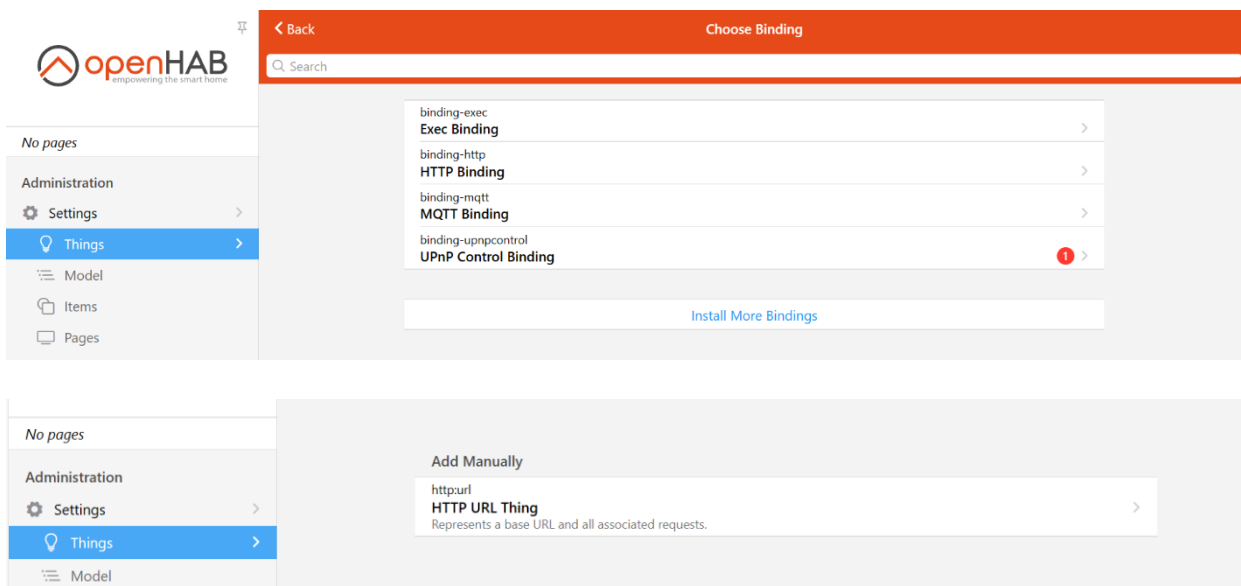
*注: 有些操作步驟和 1.2 有關, 請參考 P.12 並事先準備.

2.1 建立 things

請在左邊的 Settings 裡點選 Things, 進入 Things 的畫面後再點擊右下角的藍色圖案.



進入畫面後選擇 HTTP Binding, 再點入 HTTP URL Thing.



進入設定頁面，填入所需要配置 IP Power 的資料，請按照下方圖片指示操作：

範例型號: IP Power 9258

設定步驟:

(1) 進入 HTTP URL Thing 之後，請點擊 Show advanced.

New HTTP URL Thing

| | |
|------------|---|
| Unique ID | e9dbd477bb <small>Note: cannot be changed after the creation</small> |
| Identifier | http:urle9dbd477bb |
| Label | HTTP URL Thing |
| Location | e.g. Kitchen |

HTTP URL Thing
Represents a base URL and all associated requests.

Show advanced ☐

Base URL

Required The URL set here can be extended in the channel configuration.

Refresh Time
30
Time between two refreshes of all channels

Timeout
3000
The timeout in ms for each request

Create Thing

(2) 請依照下方圖片的圖示填入 IP9258 所需的配置資訊，以下圖案解析如下：

✓：請直接照圖片上內容複製到您的 openHAB

✗：請勿直接複製貼上，有些設定因個人所處位置設定也會不同. EX: IP 位置.

▲：此為提醒，若您沒有在裝置上改過設定，可以直接複製圖片上的內容. EX: 帳號和密碼

HTTP URL 設定:

***注意:** 若想長期穩定連線 IP Power, 建議先在 IP Power 裝置上更改連線方式為””
“固定 IP”, 以避免 IP Power 裝置自動變更 IP.

New HTTP URL Thing

Unique ID

aa77f71ce4

Note: cannot be changed after the creation

Identifier

httpurl:aa77f71ce4

Label

IP9258 relay

Location

e.g. Kitchen

HTTP URL Thing

Represents a base URL and all associated requests.

Show advanced

Base URL

http://10.33.122.48

*注意

Required The URL set here can be extended in the channel configuration.

Refresh Time

30

Time between two refreshes of all channels

Timeout

3000

The timeout in ms for each request

Delay

0

Delay between to requests

Buffer Size

2048

Size of the response buffer (default 2048 kB)

Username

admin

Basic Authentication username

Password

Basic Authentication password

Authentication Mode

Basic Authentication

Preemptive Basic Authentication

Digest Authentication

New HTTP URL Thing

✓

Authentication Mode

☐ Basic Authentication

☒ Preemptive Basic Authentication

☐ Digest Authentication

✓

State Method

☒ GET

☐ POST

☐ PUT

HTTP method (GET,POST, PUT) for retrieving a status.

✓

Command Method

☒ GET

☐ POST

☐ PUT

HTTP method (GET,POST, PUT) for sending commands.

✓

Content Type

☐ application/json

☐ application/xml

☐ text/html

☒ text/plain

☐ text/xml

The MIME content type. Only used for 'POST' and 'PUT'.

Fallback Encoding

Fallback Encoding text received by this thing's channels.

Headers

Additional headers send along with the request

Ignore SSL Errors

If set to true ignores invalid SSL certificate errors. This is potentially dangerous.

Create Thing

建立完成後, 9258 的 Things 將會顯示在 Alphabetical. 請點擊下方的 IP9258 relay 完成剩下的設定.

Things

D))

1 Things

Alphabetical

By binding

By location

1

IP9258 relay

http://url:aa77f71ce4

ONLINE

>

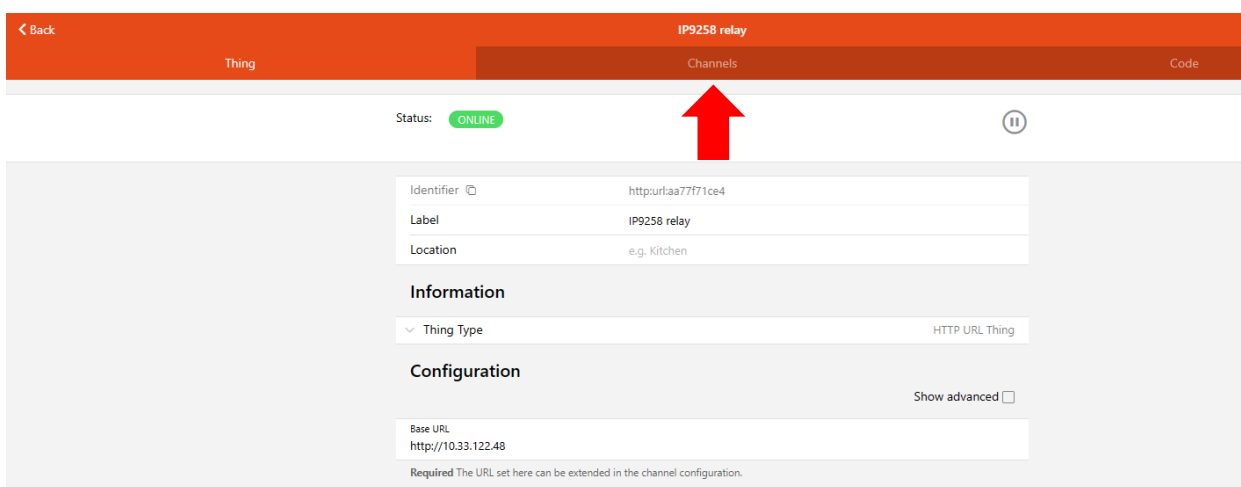
2.2 建立 Channel

9258 有 4 個 relay 開關，所以我們要創建 4 個 Channels。為了方便設定，我們準備了 4 個 map 檔案（9258_relay1.map~ 9258_relay4.map）作為指令的開關映射。這些檔案要保存在安裝目錄的.\conf\transform 下。例如：D:\openhab-4.1.2\conf\transform。

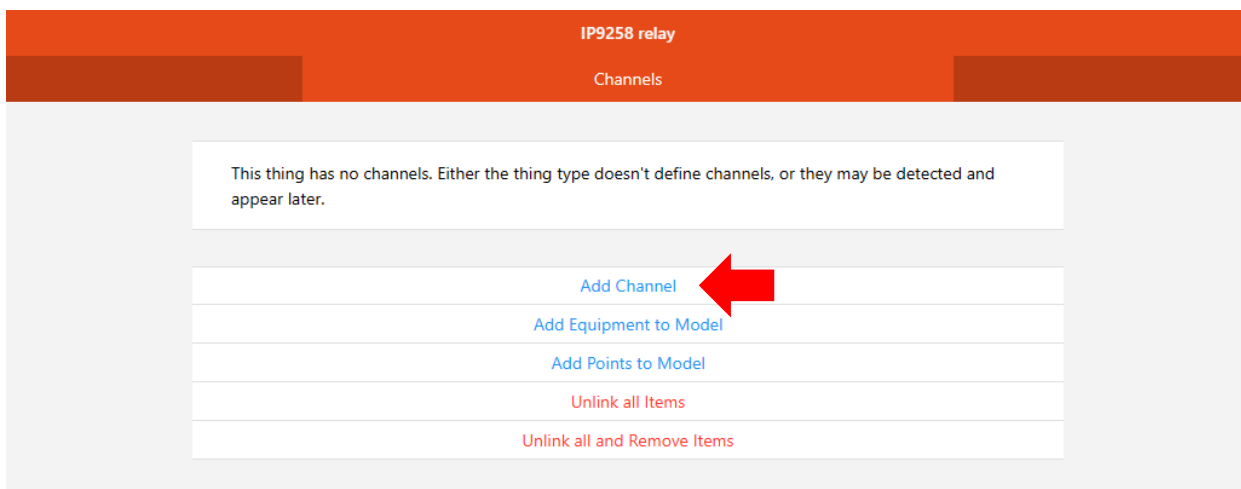
這些檔案放在 openHAB Channel 的資料夾。請將指定機型的.map 檔案放上 openHAB 的資料夾。

***注：**若您要設置 9255 系列，只需改 9258_relay1.map 的名子為 9255_relay1.map。或複製檔案再改名放入 transform 即可。

當你點進 IP9258 relay 的 Thing 時，點選圖片中紅色箭頭標示的 Channel。



再點選 Add Channel,



進入 **Add Channel** 之後，請按照下圖指示操作：

注 1: 可根據個人喜好更改名稱，但建議第一次設定先遵循下圖指示模仿。

Add Channel
IP9258 relay

Channel

注 1

▲

▲

Channel Identifier

id_IP9258_relay1

Note: cannot be changed after the creation

Label

IP9258 relay1

✕

Description

Channel type

☐ Color Channel

☐ Contact Channel

☐ DateTime Channel

☐ Dimmer Channel

☐ Image Channel

☐ Location Channel

☐ Number Channel

☐ Player Channel

☐ Rollershutter Channel

☐ String Channel

☒ Switch Channel

Add Channel
IP9258 relay

☒ Switch Channel

Configuration

Show advanced ☒

✓

State Transformation

REGEX:^(.*)p61=(d+).*

✕

Transformation pattern used when receiving values. Chain multiple transformations with the mathematical intersection character "&".

✓

Command Transformation

MAP:9258_relay1.map

✕

Transformation pattern used when sending values. Chain multiple transformations with the mathematical intersection character "&".

✓

State URL Extension

set.cmd?cmd=getpower

✕

This value is added to the base URL configured in the thing for retrieving values.

✓

Command URL Extension

set.cmd?cmd=setpower+%2\$s

✕

This value is added to the base URL configured in the thing for sending values.

全部設定完成後, 請點擊下方的 **Create**.

Escaped URL ☐

This specifies whether the URL is already escaped. Applies to the base URL, commandExtension and stateExtension.

State Content

Content for state request (only used if method is POST/PUT)

On Value
1
Required The value that represents ON

Off Value
0
Required The value that represents OFF

Read/Write Mode

☒ Read/Write
☐ Read Only
☐ Write Only

Create

建立完成後如下圖, 您的第一個 **Channel** 已完成

IP9258 relay

Thing Channels Code

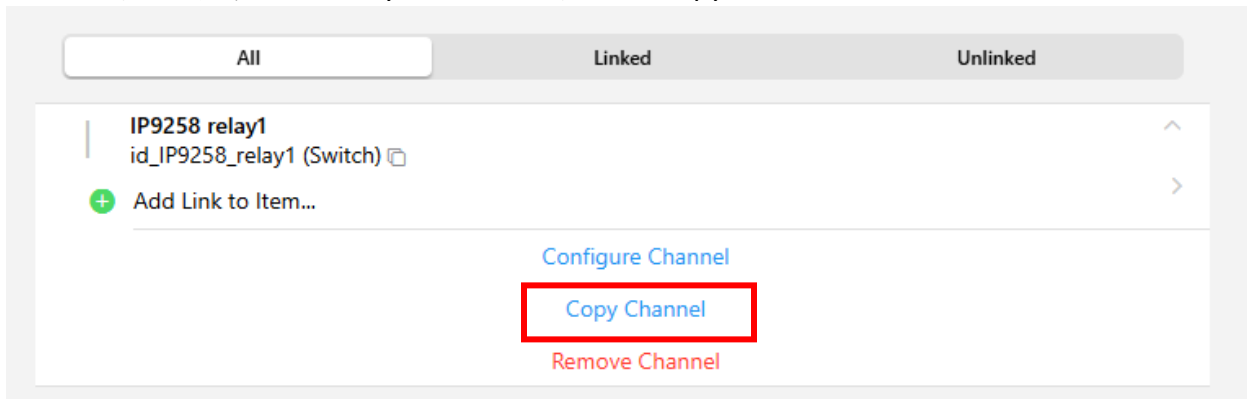
Search channels

All Linked Unlinked

IP9258 relay1
id_IP9258_relay1 (Switch)

Add Channel
Add Equipment to Model
Add Points to Model
Unlink all Items
Unlink all and Remove Items

若要另外建立其他三個 relay，只要點擊下方的 Copy Channel 即可。



以創建第二個 relay 作為範例：

點擊 Copy Channel 後，請更改圖片下方紅框裡的 code 名稱。

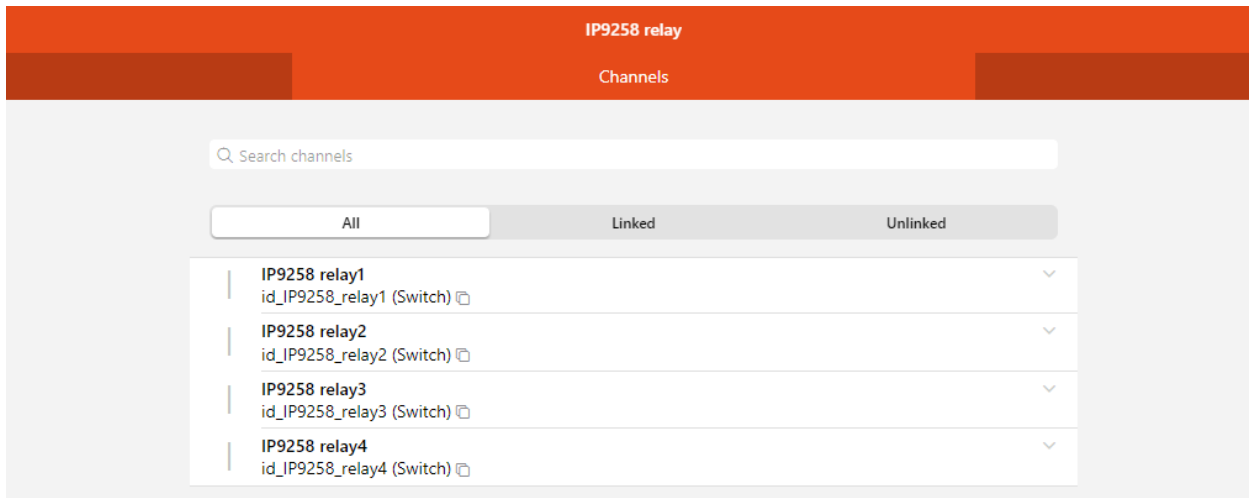
其餘設定不用更改，更改完後請按 Save。

(1) Channel Identifier: `id_IP9258_relay1_copy` → `id_IP9258_relay2`
Label: `IP9258 relay1 copy` → `IP9258 relay2`

(2) State Transformation: `REGEX: ^.*p61=(\d+).*` → `REGEX: ^.*p62=(\d+).*`
Command Transformation: `MAP:9258_relay1.map` → `MAP:9258_relay2.map`

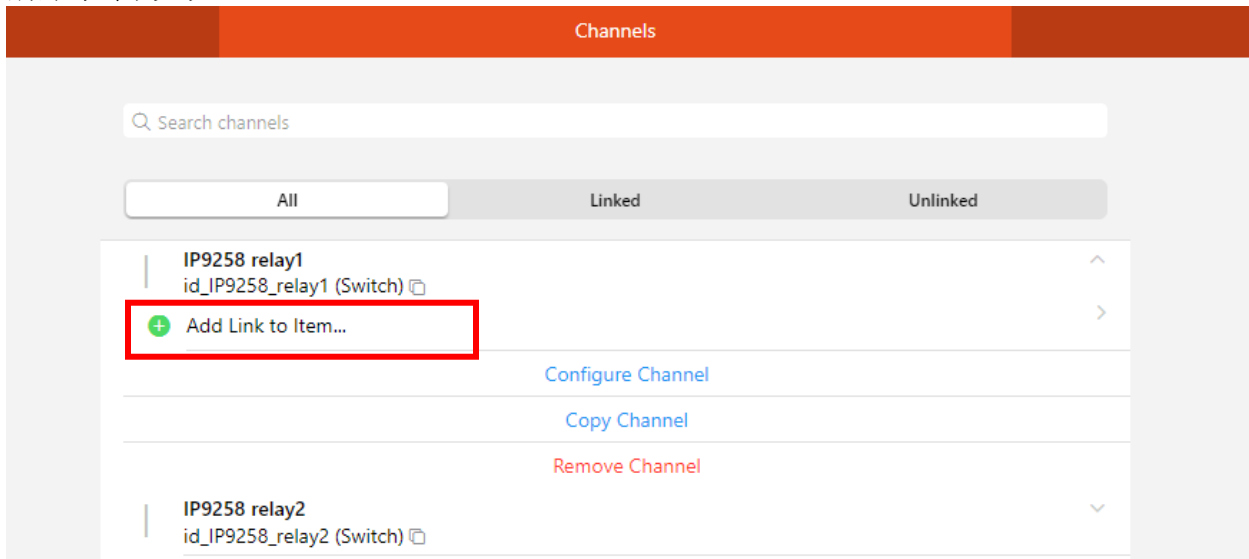
The screenshot shows the 'Copy channel' configuration page for 'IP9258 relay'. The page has an orange header with the text 'Copy channel' and 'IP9258 relay'. Below the header, there are two main sections: 'Channel' and 'Configuration'.
In the 'Channel' section, there is a red box around the 'Channel Identifier' and 'Label' fields. The 'Channel Identifier' field contains the text 'id_IP9258_relay1_copy' and a note 'Note: cannot be changed after the creation'. The 'Label' field contains the text 'IP9258 relay1 copy'.
In the 'Configuration' section, there is a red box around the 'State Transformation' and 'Command Transformation' fields. The 'State Transformation' field contains the text 'REGEX: ^.*p61=(\d+).*' and a note 'Transformation pattern used when receiving values. Chain multiple transformations with the mathematical intersection character "&".'. The 'Command Transformation' field contains the text 'MAP:9258_relay1.map' and a note 'Transformation pattern used when sending values. Chain multiple transformations with the mathematical intersection character "&".'.
Below the 'Configuration' section, there are fields for 'On Value' (containing '1') and 'Off Value' (containing '0').

完成 9258 的 Channels 後顯示如圖：



2.3 創建 Item

回到 IP9258 的 channel，我們要建立 4 個 item，我們以 IP9258 relay 1 為例，請點擊下方的“Add Link to Item...”



請填選設定與下面圖片一致：

Link Channel to Item

Channel

IP9258 relay1
http:url:aa77f71ce4:id_IP9258_relay1 (Switch)

Item

☐ Use an existing Item

☒ Create a new Item

✓

Name

IP9258_relay1

Note: cannot be changed after the creation

✕

✓

Label

IP9258 relay1

✕

Type

Switch >

Category

temperature, firstfloor...

Semantic Class

Point >

Semantic Property

None >

Non-Semantic Tags

▼

設定完成後點選“Link”，便完成設定。

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
[Learn more about profiles.](#)

✓

☒ 默认

☐ JINJA

☐ JSONPATH

☐ MAP

☐ REGEX

☐ SCRIPT ECMAScript (ECMAScript 262 Edition 11)

☐ SCRIPT Rule DSL (v1)

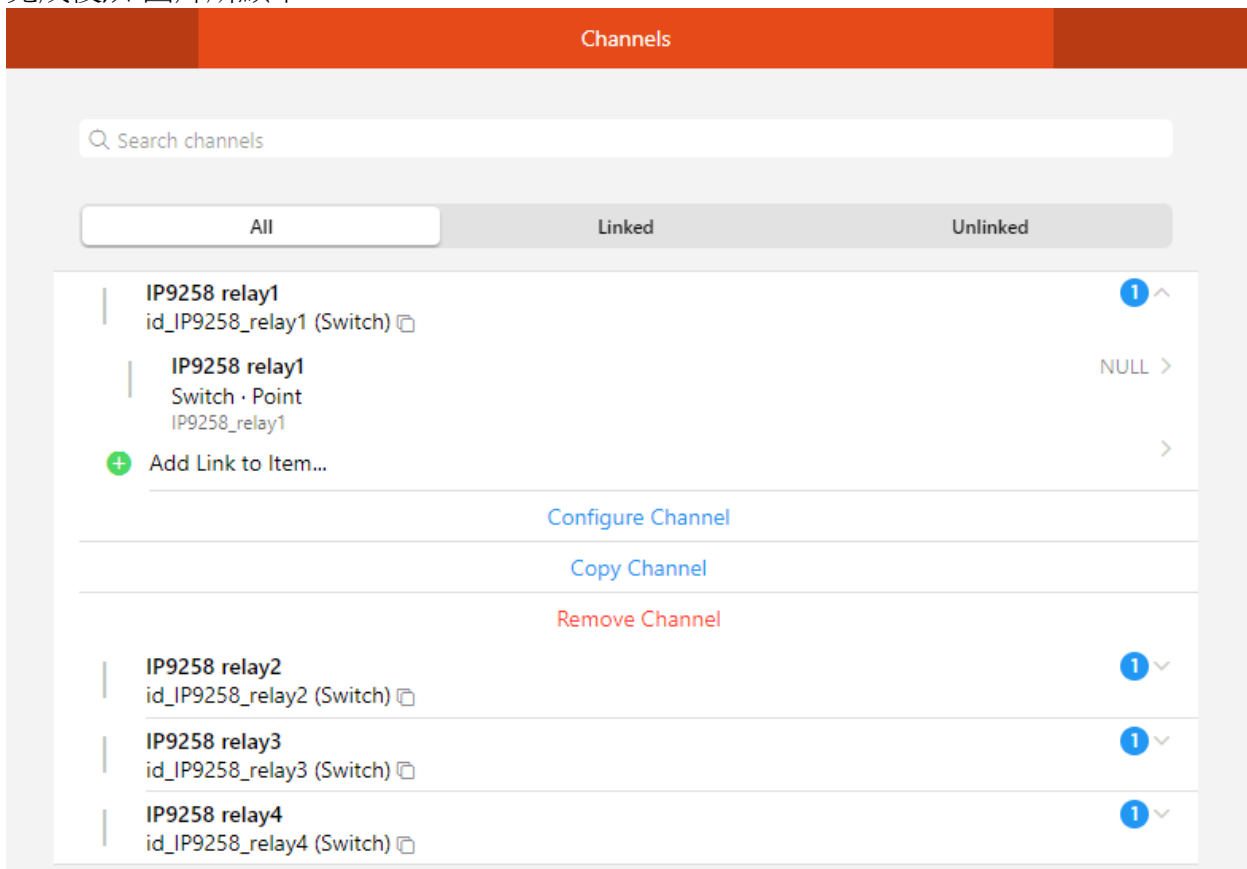
☐ 关注

☐ 更改时的时间戳

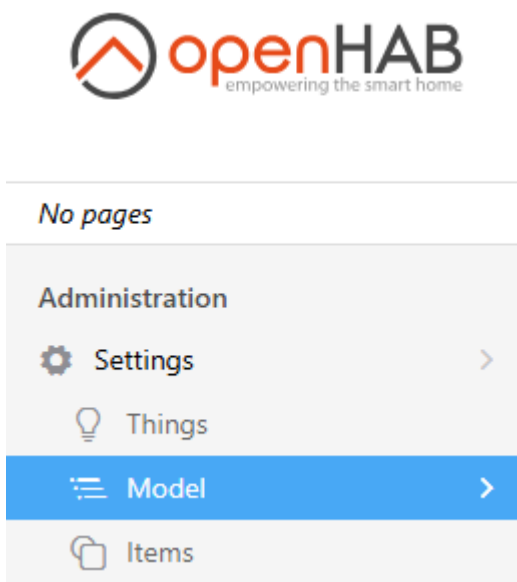
☐ 更新时的时间戳

Link

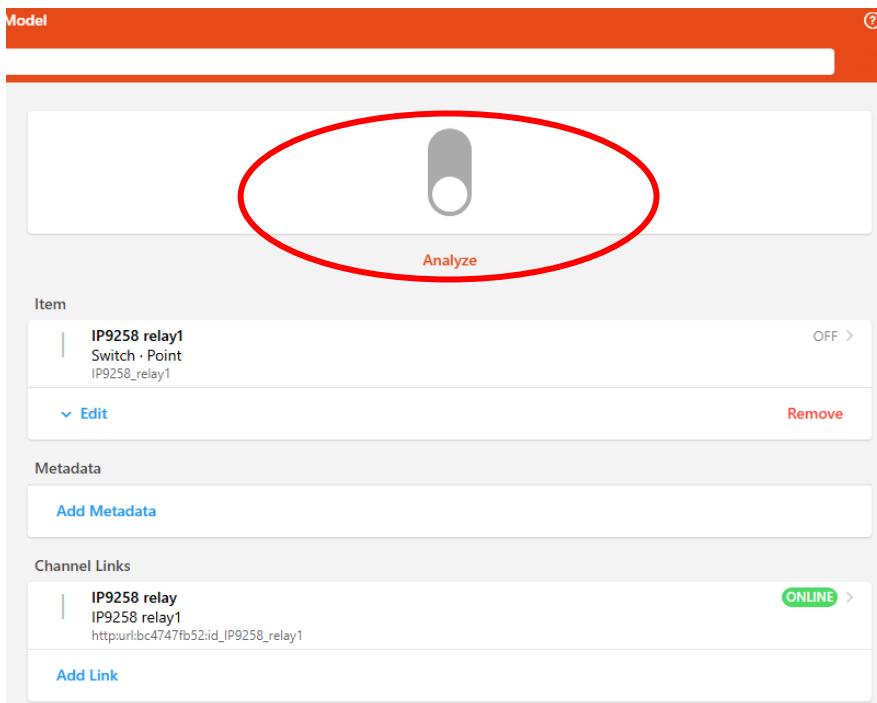
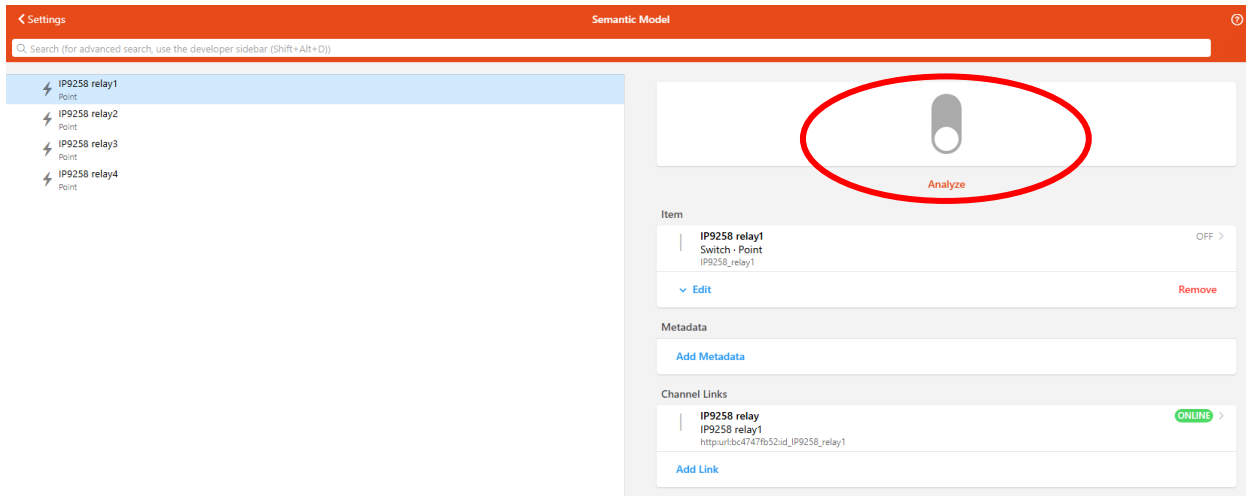
完成後,如圖片所顯示:



接著我們回到主畫面, 點選左邊的 model.



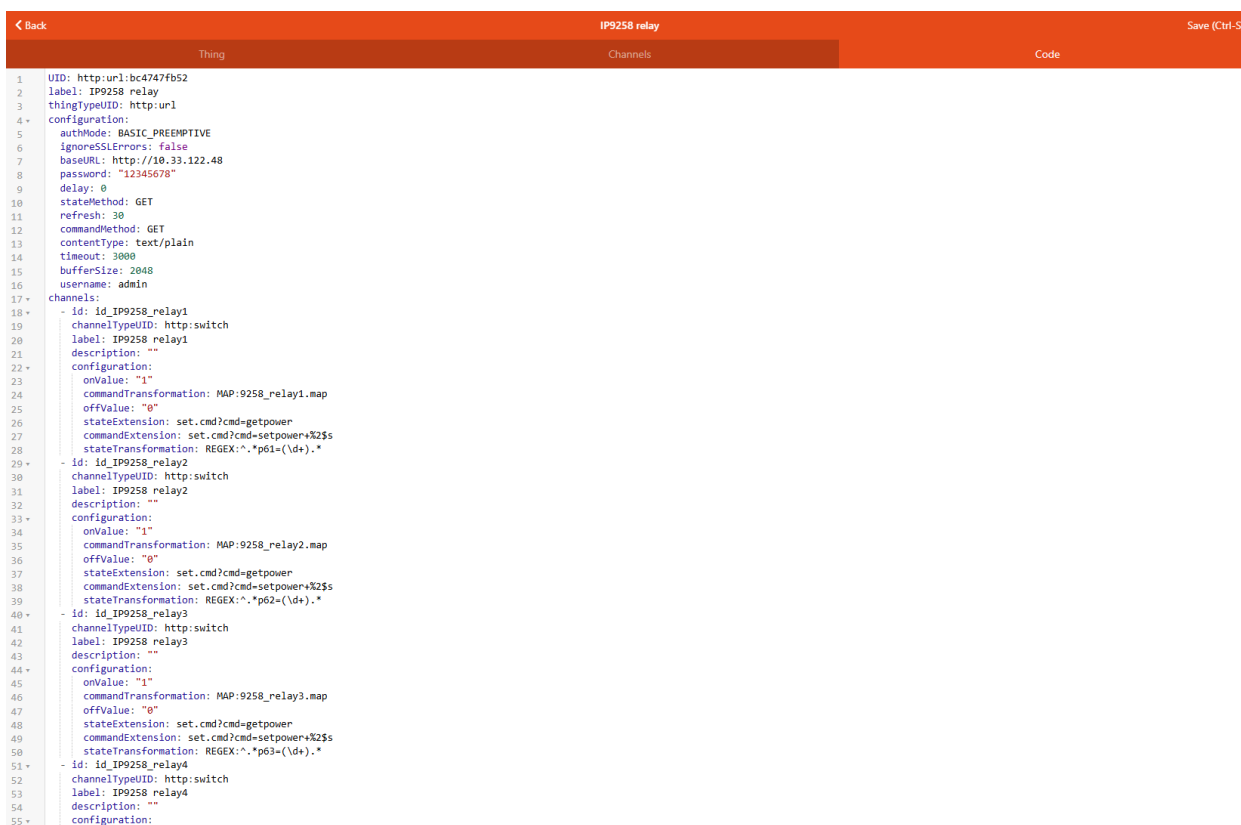
點擊之後將會進入到 Semantic Model, 近入該頁面後, 請點擊任何一個 relay, 並點擊右邊的 Switch, 若是您的 IP9258 有透過 openHAB 成功觸發, 代表您已成功建立 IP9258 在 openHAB 上的配置.



2.4 9258 系列配置參數對應的 code

4 個 channel 全部加入對應的 Item 後,透過 UI 介面配置的參數對應的程式碼可以透過 Code 頁面查看,也可以直接在 Code 頁面修改並新增新的 Channel。直接修改 code 如果出現明顯的語法錯誤,openHAB 會給予提示,修改完成記得點擊右上角的 Save (Ctrl-S) 儲存訊息,channel 和 Thing 會立刻更新。

***注:** 建議每設定 things 和 Channel 都要進入 code 檢查,因為有時候使用者新增或修改的 things 和 channel 不一定會同步設定。



```
1  UID: http:url:bc4747fb52
2  label: IP9258 relay
3  thingTypeUID: http:url
4  configuration:
5    authMode: BASIC_PREEMPTIVE
6    ignoreSSLErrors: false
7    baseUrl: http://10.33.122.48
8    password: "12345678"
9    delay: 0
10 stateMethod: GET
11 refresh: 30
12 commandMethod: GET
13 contentType: text/plain
14 timeout: 3000
15 bufferSize: 2048
16 username: admin
17 channels:
18 - id: Id_IP9258_relay1
19   channelTypeUID: http:switch
20   label: IP9258 relay1
21   description: ""
22   configuration:
23     onValue: "1"
24     commandTransformation: MAP:9258_relay1.map
25     offValue: "0"
26     stateExtension: set.cmd?cmd=getpower
27     commandExtension: set.cmd?cmd=setpower+%2$s
28     stateTransformation: REGEX:.*p61=(\d+).
29 - id: Id_IP9258_relay2
30   channelTypeUID: http:switch
31   label: IP9258 relay2
32   description: ""
33   configuration:
34     onValue: "1"
35     commandTransformation: MAP:9258_relay2.map
36     offValue: "0"
37     stateExtension: set.cmd?cmd=getpower
38     commandExtension: set.cmd?cmd=setpower+%2$s
39     stateTransformation: REGEX:.*p62=(\d+).
40 - id: Id_IP9258_relay3
41   channelTypeUID: http:switch
42   label: IP9258 relay3
43   description: ""
44   configuration:
45     onValue: "1"
46     commandTransformation: MAP:9258_relay3.map
47     offValue: "0"
48     stateExtension: set.cmd?cmd=getpower
49     commandExtension: set.cmd?cmd=setpower+%2$s
50     stateTransformation: REGEX:.*p63=(\d+).
51 - id: Id_IP9258_relay4
52   channelTypeUID: http:switch
53   label: IP9258 relay4
54   description: ""
55   configuration:
```

想參考本圖中的 Things 和 channel 設定的 code,請參考 P.51

3. 配置 IP Power 適用 9850, 9858, 9855 系列

3.1 創建 things

和 2.1 的設定方式法類似，在設定時選擇 HTTP Binding. 點擊 HTTP URL Thing，建立一個新的 http_98XX_relay 事物。

同樣在 Base URL 欄位填寫 98XX 設備的 IP 位址；Username 和 Password 欄位填寫 IP98XX 的 user 和 password；Refresh Time 欄位填寫讀取 98XX 設備開關狀態的刷新時間，這裡預設為 30 秒；State Method 選擇 GET，Command Method 選 POST；Content Type 選擇 application/json。

詳細填寫方式請參考下圖：（範例:IP POWER 9850）

The screenshot shows the configuration interface for a device named 'http_9850_relay'. The interface is organized into sections: Identifier, Label, Location, Information, Configuration, and a section for Base URL with a warning.

Identifier: http:url:9b4e47dc4d

Label: http_9850_relay

Location: e.g. Kitchen

Information: Thing Type: HTTP URL Thing

Configuration: Show advanced ☒

Base URL: http://10.33.122.52. A red 'X' icon and a warning message are present: "*在填入 IP 前, 請先參考 P.15 的注意." Below the field, it says "Required The URL set here can be extended in the channel configuration."

Refresh Time: 30. Time between two refreshes of all channels.

Timeout: 3000. The timeout in ms for each request.

Delay: 0. Delay between to requests.

Buffer Size: 2048. Size of the response buffer (default 2048 kB).

Username: admin. Basic Authentication username.

Password: Basic Authentication password.

*注: 除了 9855 系列要用 Preemptive Basic Authentication, 其他 98 系列只需點選 basic authentication 即可連線使用.

http_9850_relay

Channels

Authentication Mode

☒ Basic Authentication

☐ Preemptive Basic Authentication

☐ Digest Authentication

State Method

☒ GET

☐ POST

☐ PUT

Command Method

☐ GET

☒ POST

☐ PUT

Content Type

☒ application/json

☐ application/xml

☐ text/html

☐ text/plain

☐ text/xml

Fallback Encoding

Headers

Ignore SSL Errors

HTTP method (GET,POST, PUT) for retrieving a status.

HTTP method (GET,POST, PUT) for sending commands.

The MIME content type. Only used for 'POST' and 'PUT'.

Fallback Encoding text received by this thing's channels.

Additional headers send along with the request

If set to true ignores invalid SSL certificate errors. This is potentially dangerous.

3.2 創建 Channels

和 2.2 的操作類似，建立一個 ID 名稱為 `http_9850_relay1` 的 channel。

同樣準備 4 個 map 檔（`http_cmd1.map` ~ `http_cmd4.map`）作為對應 4 個 relay 的指令開關映射。這些檔案要保存在安裝目錄的 `.\conf\transform` 下。

例如 `http_cmd1.map` 內容如下：

```
1= {"cmd":"setpower", "RL":[{"id":1, "action":1}]}
```

```
0= {"cmd":"setpower", "RL":[{"id":1, "action":0}]}
```

JSON 格式的 "id":1 表示第一個 relay, "action":1 表示開啟。

Add Channel
http_9850_relay

Channel

✓

Channel Identifier

id_http_9850_relay1

Note: cannot be changed after the creation

✓

Label

http 9850 relay1

✕

Description

Channel type

☐ Color Channel

☐ Contact Channel

☐ DateTime Channel

☐ Dimmer Channel

☐ Image Channel

☐ Location Channel

☐ Number Channel

☐ Player Channel

☐ Rollershutter Channel

☐ String Channel

☒ Switch Channel

請根據下方的指示填入 state Transformation
JSONPATH:\$.result.RL[??.state 請參照下面範例填入:

EX:

JSONPATH:\$.result.RL[0].state → Relay 1

JSONPATH:\$.result.RL[1].state → Relay 2

JSONPATH:\$.result.RL[2].state → Relay 3

·
·
·

請依照您的機型電卓數量做調整.

Configuration

Show advanced ☒

State Transformation
JSONPATH:\$.result.RL[0].state

Transformation pattern used when receiving values. Chain multiple transformations with the mathematical intersection character "∩".

Command Transformation
MAP:http_cmd1.map

Transformation pattern used when sending values. Chain multiple transformations with the mathematical intersection character "∩".

State URL Extension
json.cmd?getpower

This value is added to the base URL configured in the thing for retrieving values.

Command URL Extension
json.cmd?

This value is added to the base URL configured in the thing for sending values.

確認與圖片中的設定配置一致後, 點擊 create, 9850 的 Channel 就此完成.

On Value
1

Required The value that represents ON

Off Value
0

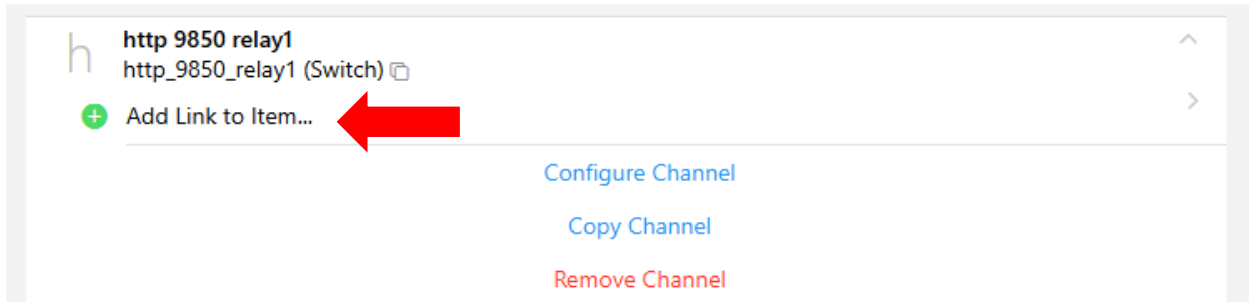
Required The value that represents OFF

Read/Write Mode
☒ Read/Write
☐ Read Only
☐ Write Only

Create

3.3 創建 item

回到 IP9850 的 channel, 建立新的 item, 建立的過程與 2.2 一樣, 確認與下圖的設定一致即可.



| | | |
|---------------------|--|---|
| ✓ Name | http_9850_relay1 | ⓧ |
| | Note: cannot be changed after the creation | |
| ✓ Label | http 9850 relay1 | ⓧ |
| ✓ Type | Switch | > |
| Category | temperature, firstfloor... | |
| | <input type="checkbox"/> | |
| ✓ Semantic Class | Point | > |
| ✓ Semantic Property | None | > |

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
[Learn more about profiles.](#)

| |
|---------------------------------------|
| ✓ <input checked="" type="radio"/> 默认 |
| <input type="radio"/> JINJA |
| <input type="radio"/> JSONPATH |

3.4 98 系列配置參數對應的 code

查詢和修改方式與 2.3 一樣, 想參考該手冊 98 系列配置 code 請參閱 P.52

3.5 創建屬性 Channel

有些 IP POWER 的型號具有偵測溫度，電流和電壓的功能，為了能偵測到這些屬性，我們需要在有這類型功能的 IP POWER 上另外建立一個 channel.

示範機型: IP POWER 9855 PRO

在建立新的 IP POWER 的 Thing 之後，在其內部建立一個 Channel.

Link Channel to Item

Channel
http 9855 PRO temperature
http:url:c826bee367:http_9855_PRO_temperature (Number)

Item
☐ Use an existing Item
☒ Create a new Item

| | | |
|-----------|--|---|
| Name | http_9855_PRO_temperature | × |
| | Note: cannot be changed after the creation | |
| Label | http 9855 PRO temperature | × |
| Type | Number | > |
| Dimension | | > |
| Category | temperature, firstfloor... | |

| | | |
|-------------------|-------|---|
| Semantic Class | Point | > |
| Semantic Property | None | > |

| | |
|-------------------|---|
| Non-Semantic Tags | ▼ |
|-------------------|---|

| | |
|-----------------|---|
| Parent Group(s) | > |
|-----------------|---|


Profile
Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
[Learn more about profiles.](#)

☒ 默认
☐ JINJA
☐ JSONPATH

除 Channel Identifier 和 Label 外, 其他設定直接照圖片上複製即可.

Add Channel
http_9855_PRO_relay

Channel

| | |
|--------------------|---|
| Channel Identifier | http_9855_PRO_temperature <small>Note: cannot be changed after the creation</small> |
| Label | http 9855 PRO temperature  |
| Description | |

Channel type

☐ Color Channel

☐ Contact Channel

☐ DateTime Channel

☐ Dimmer Channel

☐ Image Channel

☐ Location Channel

☒ Number Channel

☐ Player Channel

☐ Rollershutter Channel

☐ String Channel

☐ Switch Channel

在填完 Configuration 的設定後, 點選 create 便可完成.

*注: 在圖片中的 State Transformation 和 unit 中, 正在設定的屬性為“溫度”.

若要設定其他屬性, 請將下方 code 中的紅字變更:

Ex: JINJA:{{value_json.result.global_measure.temperature}}

例: temperature → voltage.

例: Unit: °C (temperature), AC (voltage), A (current)

*部分設定可能會因屬性不一樣而需要做變更, 請參考 P.56 (重要)

Configuration

Show advanced ☒

State Transformation

JINJA:{{value_json.result.global_measure.temperature}}

Transformation pattern used when receiving values. Chain multiple transformations with the mathematical intersection character "∩".

Command Transformation

Transformation pattern used when sending values. Chain multiple transformations with the mathematical intersection character "∩".

State URL Extension

json.cmd?getpower

This value is added to the base URL configured in the thing for retrieving values.

Command URL Extension

This value is added to the base URL configured in the thing for sending values.

Escaped URL

☐

This specifies whether the URL is already escaped. Applies to the base URL, commandExtension and stateExtension.

State Content

Content for state request (only used if method is POST/PUT)

Read/Write Mode

☐ Read/Write

☒ Read Only

☐ Write Only

Unit

°C

Unit to append to the (transformed) value.

完成 Channel 後, 再建立 item, 基本上無須做變更, 與下圖設定相符即可.

確認設定無誤後創建 item.

Link Channel to Item

Channel

http 9855 PRO temperature

http:url:c826bee367:http_9855_PRO_temperature (Number)

Item

☐ Use an existing Item

☒ Create a new Item

Name

http_9855_PRO_temperature

Note: cannot be changed after the creation

Label

http 9855 PRO temperature

Type

Number >

Dimension

>

Category

temperature, firstfloor...

Semantic Class

Point >

Semantic Property

None >

Non-Semantic Tags

>

Parent Group(s)

>

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
[Learn more about profiles.](#)

☒ 默认

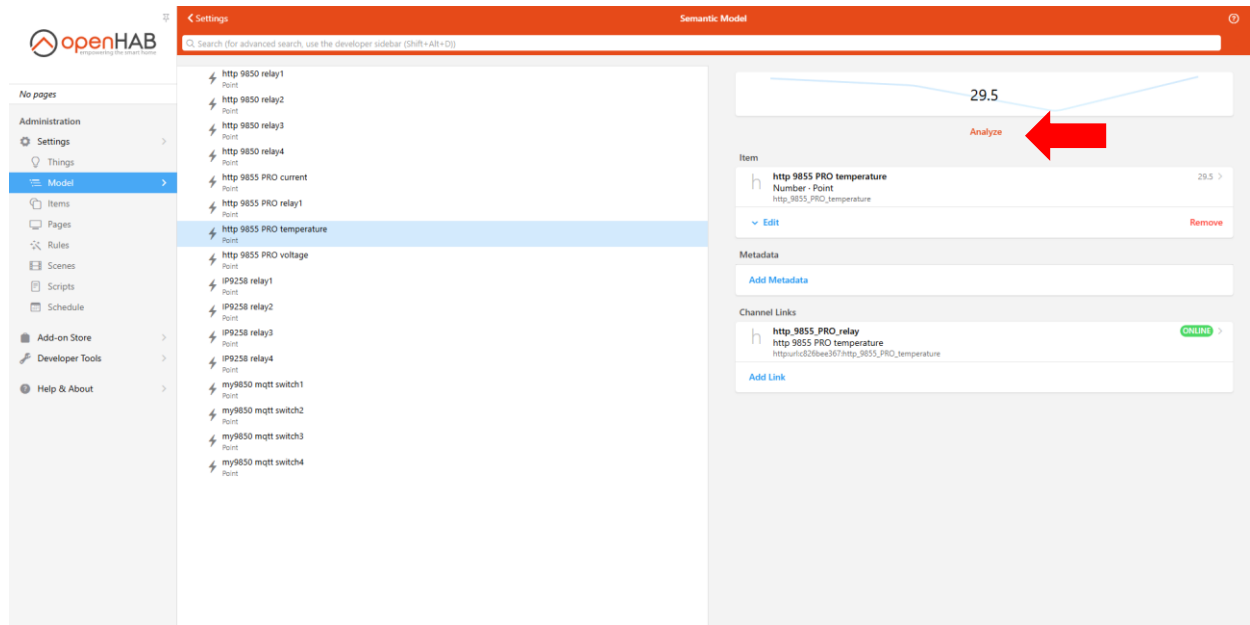
☐ JINJA

☐ JSONPATH

☐ MAP

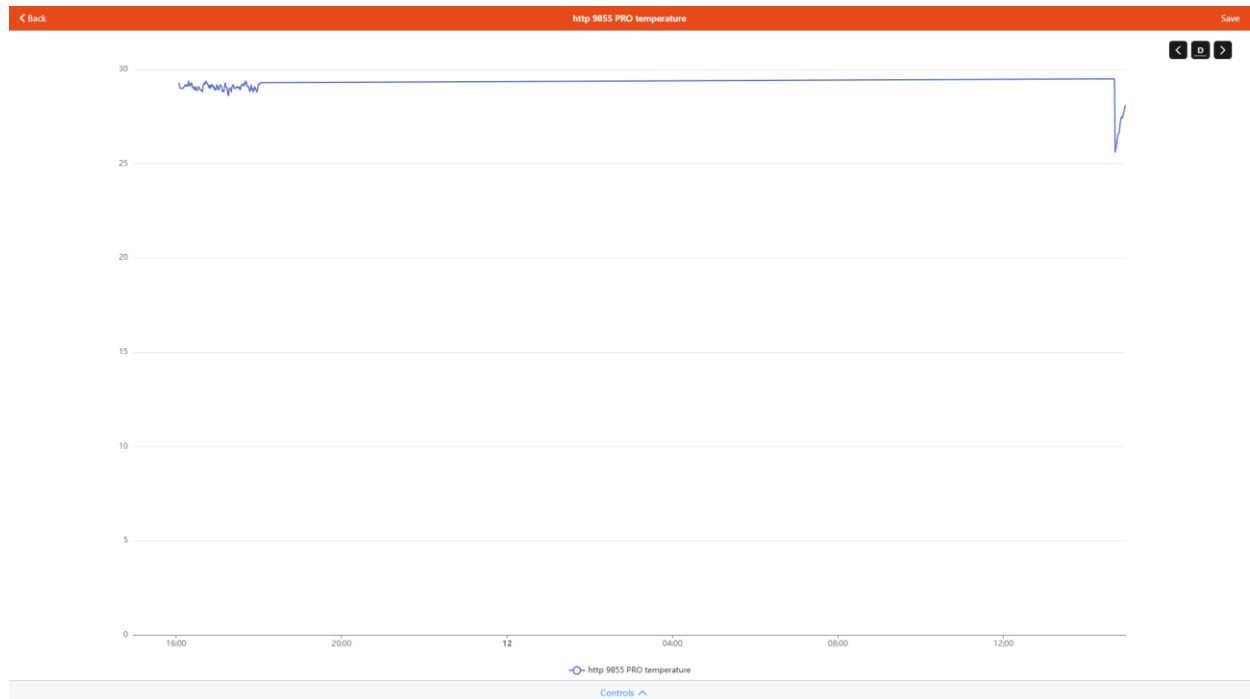
完成設定之後, 可在 model 中顯示, 點擊右邊 Analyze 可顯示圖表.

主頁面:



The screenshot shows the openHAB Semantic Model interface. The left sidebar contains a menu with 'Model' selected. The main area lists various items, including 'http 9855 PRO temperature'. On the right, the 'Analyze' button is highlighted with a red arrow, and the current value '29.5' is displayed above it.

溫度圖表:

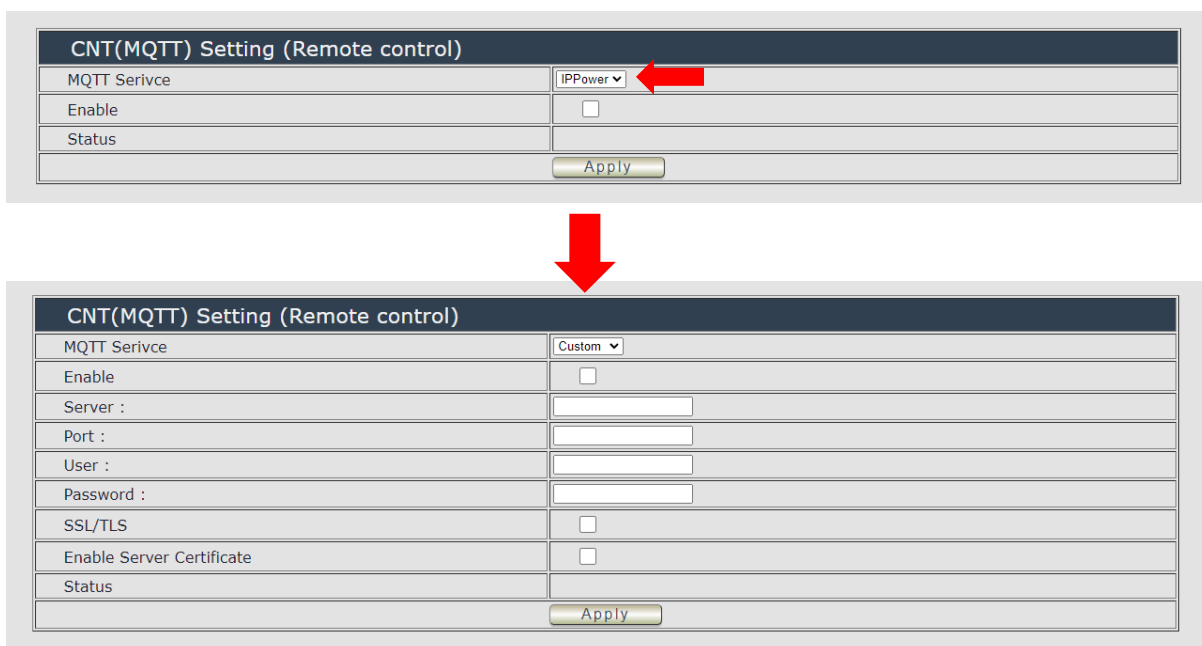


4. MQTT 設定

此方式能讓使用者連結和控制內網以外的 IP Power 裝置，請依照下列步驟：
(目前此功能僅支援 IP Power 98 系列)

(4-1) 首先，請開啟你的 IP POWER 裝置，並進入 Application→IP Service.

選擇 CNT(MQTT) Setting，並點擊 MQTT Service，選擇 Custom.



The image shows two screenshots of the 'CNT(MQTT) Setting (Remote control)' interface. The top screenshot shows the 'MQTT Service' dropdown menu set to 'IPPower', with a red arrow pointing to it. The bottom screenshot shows the 'MQTT Service' dropdown menu set to 'Custom', with a red arrow pointing to it. The bottom screenshot also shows additional fields for 'Server', 'Port', 'User', 'Password', 'SSL/TLS', and 'Enable Server Certificate'.

| CNT(MQTT) Setting (Remote control) | |
|------------------------------------|--------------------------|
| MQTT Service | IPPower |
| Enable | <input type="checkbox"/> |
| Status | |
| Apply | |

| CNT(MQTT) Setting (Remote control) | |
|------------------------------------|--------------------------|
| MQTT Service | Custom |
| Enable | <input type="checkbox"/> |
| Server : | |
| Port : | |
| User : | |
| Password : | |
| SSL/TLS | <input type="checkbox"/> |
| Enable Server Certificate | <input type="checkbox"/> |
| Status | |
| Apply | |

(4-2) 若您已有專屬的 MQTT broker，請填寫上方資訊，
SSL/TLS 和認證(Certificate)依使用者需求設定，但 Enable 請務必開啟。
設定完後請按 “Apply”，按完後若 Status 出現 “Connect” 代表成功啟用。

若您沒有自己的 MQTT broker，可以選擇一些線上免費提供的 MQTT broker。
例如：broker.emqx.io，請不要選擇 test.mosquitto.org，因為和我們的設備有相容性問題。
選擇公用的 broker 請盡量使用 8883 端口的 TLS 安全連線，因為會暴露在所有集成了這個 mqtt broker 的平台面前，所以請注意安全風險。

*此次教學會以 1883 端口作為設定範例。

*若想使用本公司的 IP POWER 作為 MQTT Broker，請聯繫我們。

4.1 創建 MQTT Broker

點選 MQTT Binding → 再點選 MQTT Broker.

| | | |
|--------------|---------------------|---|
| binding-exec | Exec Binding | > |
| binding-http | HTTP Binding | > |
| binding-mqtt | MQTT Binding | > |

Add Manually

| | | |
|--|---------------------------|----------|
| mqtt:broker | MQTT Broker | Bridge > |
| A connection to a MQTT broker | | |
| mqtt:topic | Generic MQTT Thing | > |
| You need a configured Broker first. Dynamically add channels of various types to this Thing. Link different MQTT topics to each channel. | | |

請按照下方圖示設定, 若您是自己的 broker 做設定, 請更改 Broker Hostname/IP.

| | |
|------------|------------------------|
| Identifier | mqtt:broker:8496c99244 |
| Label | MQTT Broker |
| Location | e.g. Kitchen |

MQTT Broker

A connection to a MQTT broker

Show advanced ☒

Broker Hostname/IP

broker.emqx.io

Required The IP/Hostname of the MQTT broker

Broker Port

1883

The port is optional, if none is provided, the typical ports 1883 and 8883 (SSL) are used.

Secure Connection

☐


Required Uses TLS/SSL to establish a secure connection to the broker.

Hostname Validated

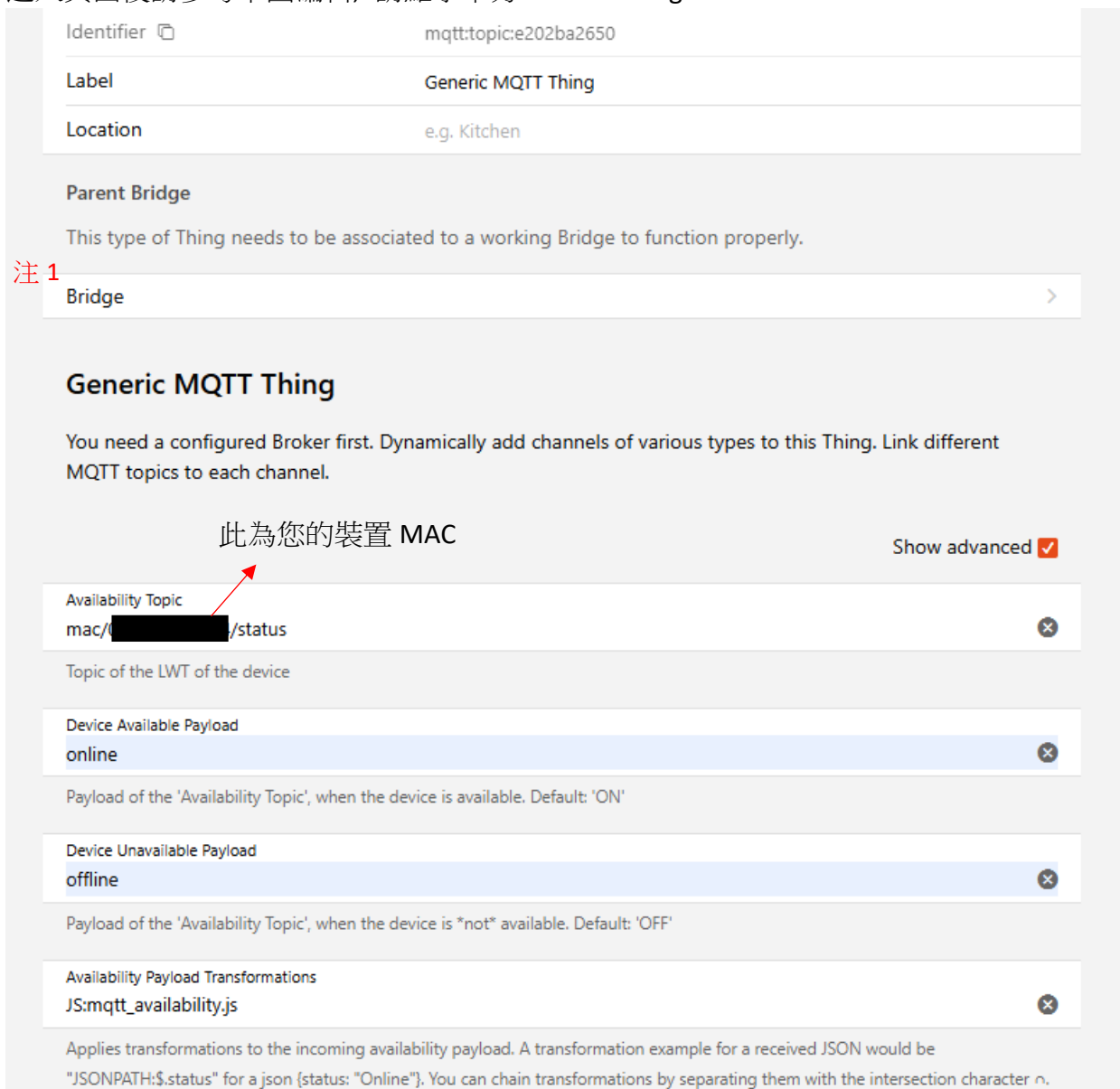
☒


Validate hostname from certificate against server hostname for secure connection.

4.2 創建 Generic MQTT Thing

與 4.12 的建立方式一樣，在 things 的頁面點擊  再點選 MQTT Binding → Generic MQTT Thing.

進入頁面後請參考下圖編輯，請點擊下方 Create Thing.



Identifier  mqtt:topic:e202ba2650

Label Generic MQTT Thing

Location e.g. Kitchen

Parent Bridge

This type of Thing needs to be associated to a working Bridge to function properly.


注 1 Bridge >

Generic MQTT Thing


You need a configured Broker first. Dynamically add channels of various types to this Thing. Link different MQTT topics to each channel.

此為您的裝置 MAC


Show advanced ☒

Availability Topic
mac/[redacted]/status 


Topic of the LWT of the device

Device Available Payload
online 

Payload of the 'Availability Topic', when the device is available. Default: 'ON'

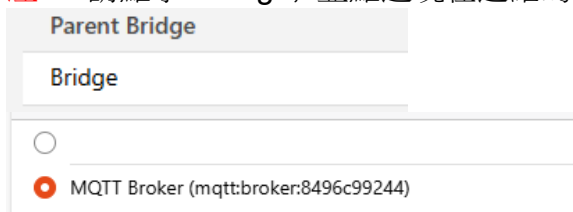
Device Unavailable Payload
offline 

Payload of the 'Availability Topic', when the device is *not* available. Default: 'OFF'

Availability Payload Transformations
JS:mqtt_availability.js 


Applies transformations to the incoming availability payload. A transformation example for a received JSON would be "JSONPATH:\$.status" for a json {status: "Online"}. You can chain transformations by separating them with the intersection character ^.

注 1: 請點擊 Bridge，並點選現在連結的 MQTT Broker.



Parent Bridge

Bridge

☐ 


☒ MQTT Broker (mqtt:broker:8496c99244)

4.3 創建 Generic MQTT Thing 的 Channels

請參考下方圖片中的設定創建 Channels.

若您是第一次設定, Channel 裡的 Identifier 和 Label 名稱除 IP Power 以外, 其他文字建議不要更動, 與 P.XX 對照 Code 會比較方便.

Channel

| | |
|--------------------|---|
| Channel Identifier | my9850_mqtt_switch1 |
| | Note: cannot be changed after the creation |
| Label | my9850 mqtt switch1  |
| Description | |

Channel type

☐ Text Value

☐ Number Value

☐ Dimmer

☒ On/Off Switch

☐ Open/Close Contact

☐ Color Value (Red,Green,Blue)

☐ Color Value (Hue,Saturation,Brightness)

☐ Color Value (HSB, RGB or CIE xyY)

☐ Date/Time Value
Current date and/or time

☐ Image
An image to display. Send a binary bmp, jpg, png or any other supported format to this channel.

☐ Location
GPS coordinates as Latitude,Longitude,Altitude

☐ Rollershutter

☐ Trigger

Advanced 設定 1

注 1:被遮住的部分是您要填入的裝置 MAC.

Configuration

Show advanced ☒

MQTT State Topic 注 1

mac/[REDACTED]/cmd_ack/id_my_openhab

An MQTT topic that this thing will subscribe to, to receive the state. This can be left empty, the channel will be state-less command-only channel.

MQTT Command Topic 注 1

mac/[REDACTED]/cmd

An MQTT topic that this thing will send a command to. If not set, this will be a read-only switch.

QoS

☒ At most once (best effort delivery "fire and forget")

☐ At least once (guaranteed that a message will be delivered at least once)

☐ Exactly once (guarantees that each message is received only once by the counterpart)

MQTT QoS of this channel (0, 1, 2). Default is QoS of the broker connection.

Retained

☐

The value will be published to the command topic as retained message. A retained value stays on the broker and can even be seen by MQTT clients that are subscribing at a later point in time.

Is Command

☐

If the received MQTT value should not only update the state of linked items, but command them, enable this option.

Custom On/Open Value

p1_1

A number (like 1, 10) or a string (like "enabled") that is additionally recognised as on/open state. You can use this parameter for a second keyword, next to ON (OPEN respectively on a Contact).

Custom Off/Closed Value

p1_0

A number (like 0, -10) or a string (like "disabled") that is additionally recognised as off/closed state. You can use this parameter for a second keyword, next to OFF (CLOSED respectively on a Contact).

Advanced 設定 2

請注意 Incoming Value Transformations 的部分，
由於此段要輸入的 Script 較長，請複製下方文字段落。

Incoming Value Transformations Script:

```
JINJA:"{% if (value_json.cmd == 'getpower' and value_json.result.RL[1].state == 1) or  
(value_json.tag == 'r1_on' and value_json.result.message == 'success') %} 'p1_1' {% elif  
(value_json.cmd == 'getpower' and value_json.result.RL[1].state == 0) or (value_json.tag ==  
'r1_off' and value_json.result.message == 'success')%} 'p1_0' {% else %} 'p1_0' {% endif %}"
```

後面若要再創建一樣的 Channels，請修改紅字部分，

Ex:

1. r1_on → r2_on
2. P1_1 → P2_2
3. r1_off → r2_off
4. P1_0 → P2_0

Transform Values

These configuration parameters allow you to alter a value before it is published to MQTT or before a received value is assigned to an item.

Incoming Value Transformations

JINJA:"{% if (value_json.cmd == 'getpower' and value_json.result.RL[0].state == 1) or (value_json.tag == 'r1_on' and

Applies transformations to an incoming MQTT topic value. A transformation example for a received JSON would be "JSONPATH:\$device.status.temperature" for a json {device: {status: { temperature: 23.2 }}}. You can chain transformations by separating them with the intersection character &.

Outgoing Value Transformation

MAP:mqtt_cmd.map

Applies a transformation before publishing a MQTT topic value. Transformations are specialised in extracting a value, but some transformations like the MAP one could be useful.

Outgoing Value Format

%s

Format a value before it is published to the MQTT broker. The default is to just pass the channel/item state. If you want to apply a prefix, say "MYCOLOR," you would use "MYCOLOR,%s". If you want to adjust the precision of a number to for example 4 digits, you would use "%.4f".

確認圖片中的設定全部輸入完成後，按下 **Create** 便能建立完成。

4.4 建立 items

若您已完成前面的步驟，為前面的 **relay channel** 指令 **channel** 建立 **Item**。
點選這些 **channel** 下面的 **Add Link to Item**,請根據您建立的 **Channel** 數量建立 **Item**。

除了圖片上的 **IP Power** 型號，其他設定都不用變更，直接點擊 **Link** 可完成。

Channel

my9850 mqtt switch1
mqtt:topic:e202ba2650:my9850_mqtt_switch1 (Switch)

Item

☐ Use an existing Item

☒ Create a new Item

Name

Generic_MQTT_Thing_my9850_mqtt_switch1

Note: cannot be changed after the creation

Label

my9850 mqtt switch1

Type

Switch >

Category

temperature, firstfloor...

Semantic Class

Point >

Semantic Property

None >

Non-Semantic Tags

>

Parent Group(s)

>

Profile

Profiles define how Channels and Items work together. Install transformation add-ons to get additional profiles.
[Learn more about profiles.](#)

☒ 默认

☐ JINJA

☐ JSONPATH

☐ MAP

☐ REGEX

☐ SCRIPT ECMAScript (ECMAScript 262 Edition 11)

4.5 MQTT 的 Code

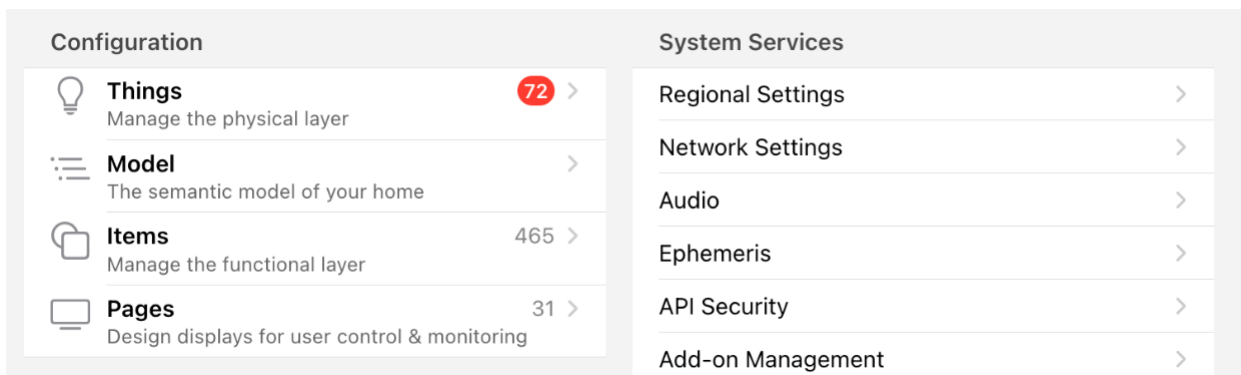
查詢和修改方式與 2.3 一樣，想參考 **MQTT 配置 code** 請參考下方 **P.54**

5. openHAB App 內網(LAN)和外網(WAN)設定:

openHAB 有自己專屬的 App 可以操控裝置, 但該 App 平時只能支援內網控制.

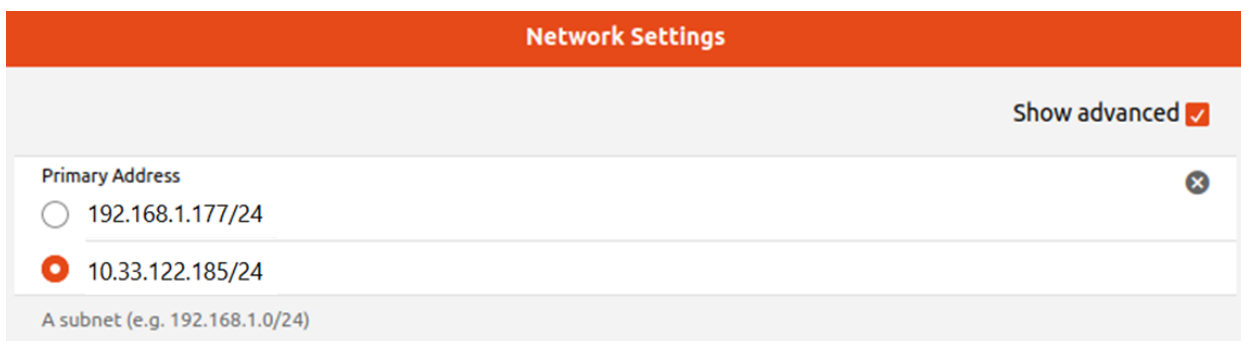
若想用手機或平板在外面透過 App 操控家裡/工作場所的 IP Power, 請依照下方步驟:

- (1) 請先到 openHAB 的 Settings → System Services
點選下方的 Network Settings.



點進去之後您會看到主要位置(Primary Address), 請選擇一個您要設置的 IP.
點選完畢後點擊右上方儲存.

*注: 若要使用 APP 操控裝置,此為必要步驟. 因為剛創建的 openHAB 是沒有預設 IP 的.

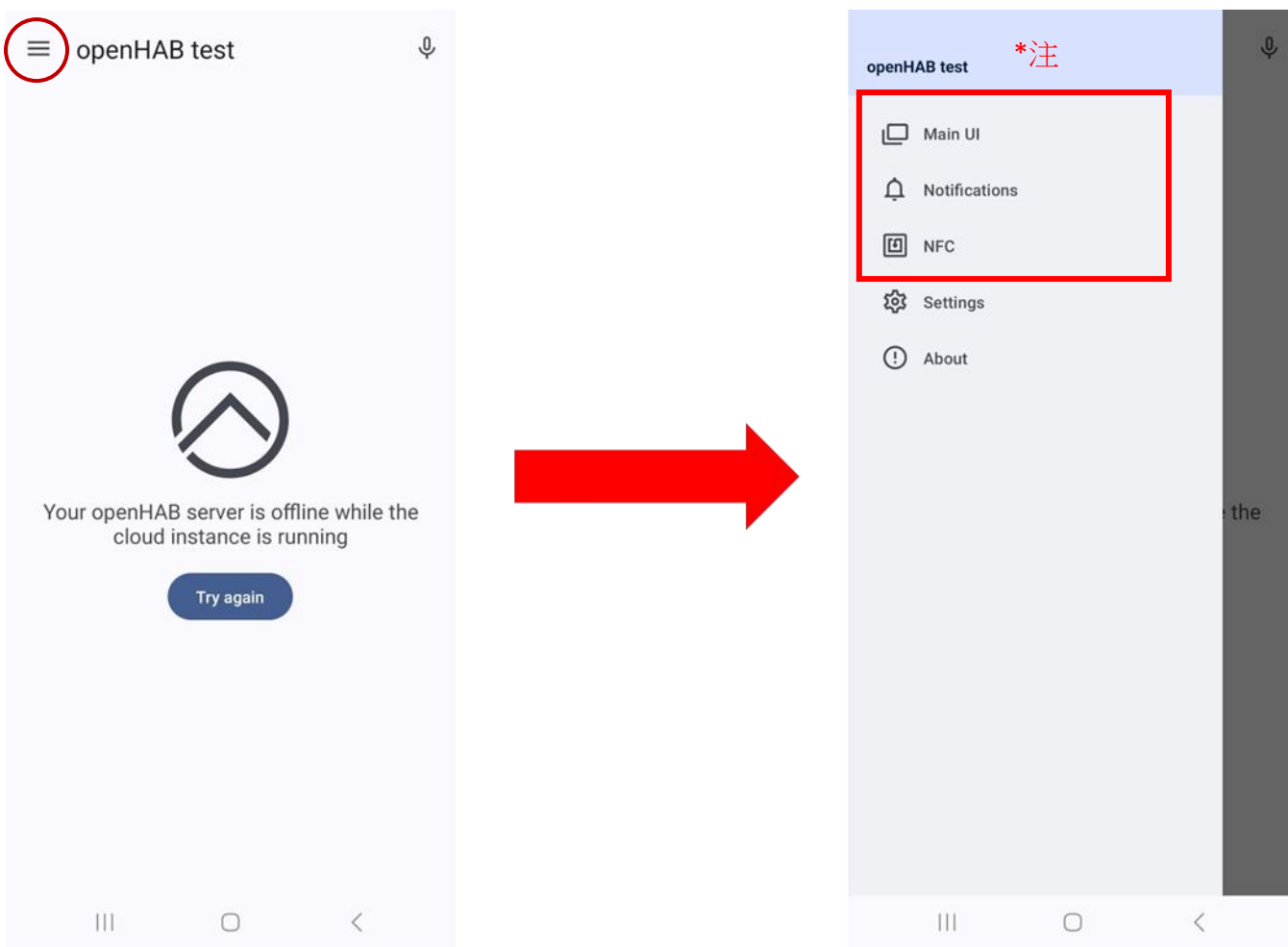


- (2) 設定完成後, 請用手機或平板下載 openHAB 的 App.

(3) 下載完成後，請打開 openHAB 手機 App，並點選下面左邊圖片中紅圈裡的圖案。
畫面出現右邊圖片的畫面後，點選 settings.

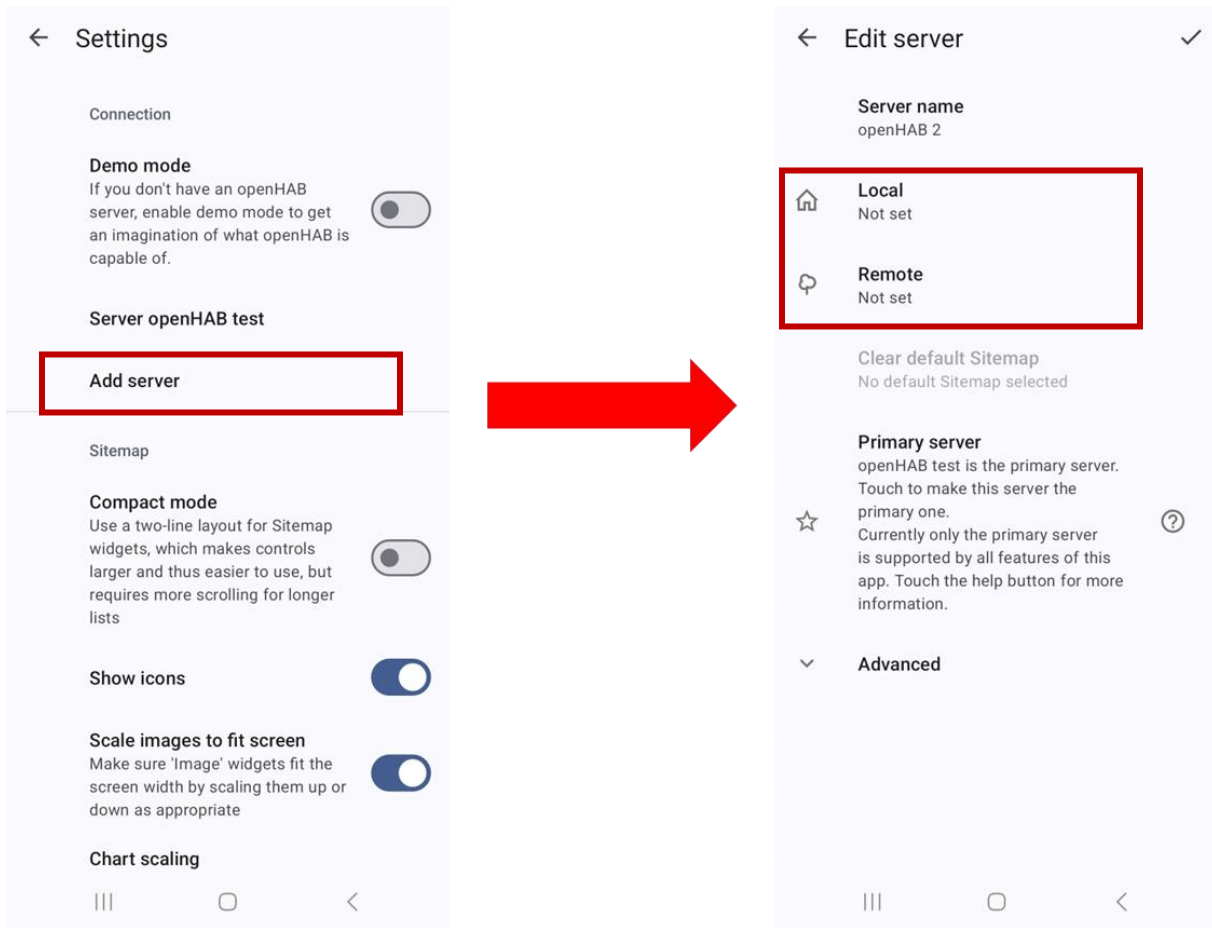
***注：**第一次開啟 openHAB 時是不會出現紅框內的選項的.

圖片範例：



(4) 進入 Settings 後, 點擊 Add server:

*若要設定內網, 請點擊 Local (建議優先)
設定外網, 請點擊 Remote.



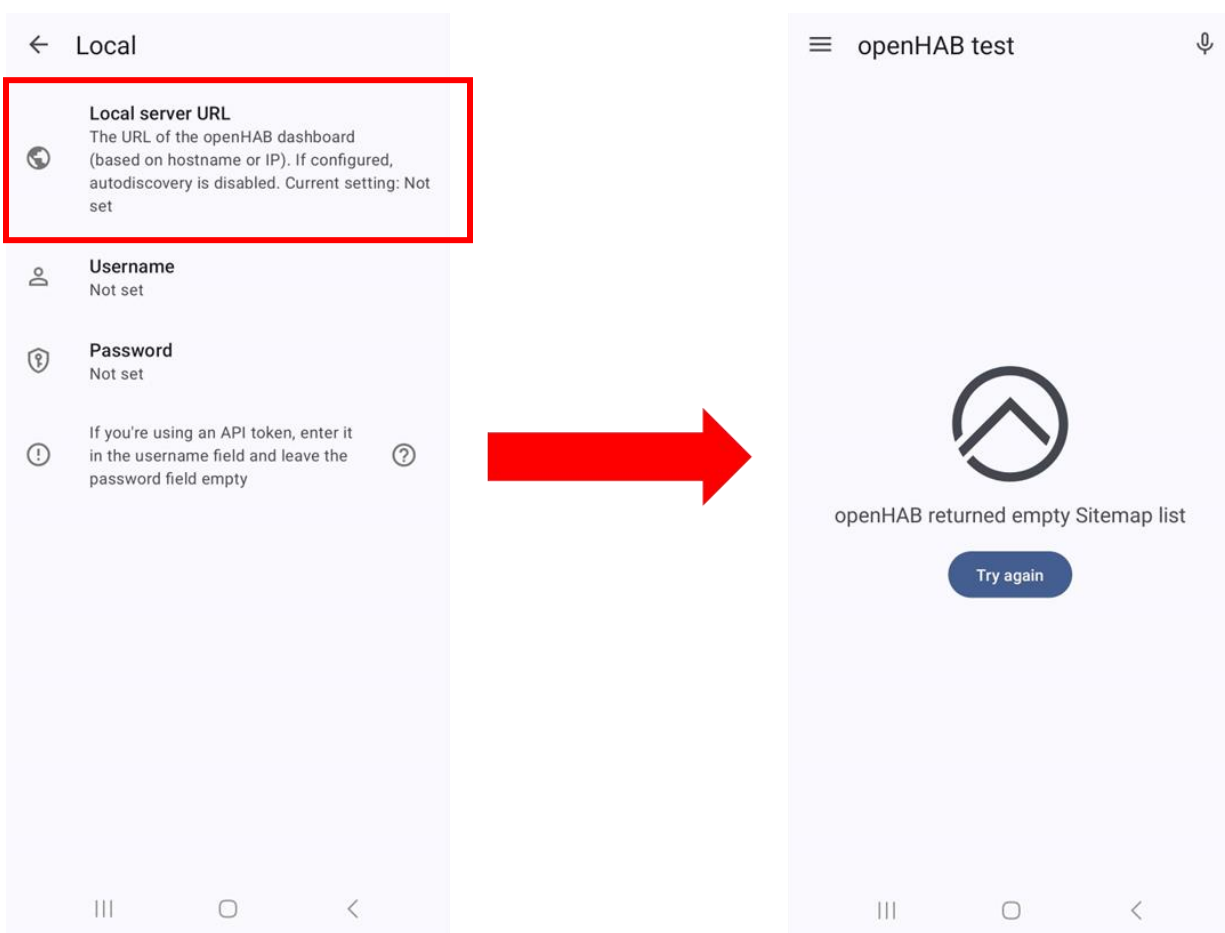
(5) 設定內網(Local)教學:

請點擊圖片紅框裡的 **Local server URL**, 並輸入您在 openHAB 中設定的主要位置(IP).

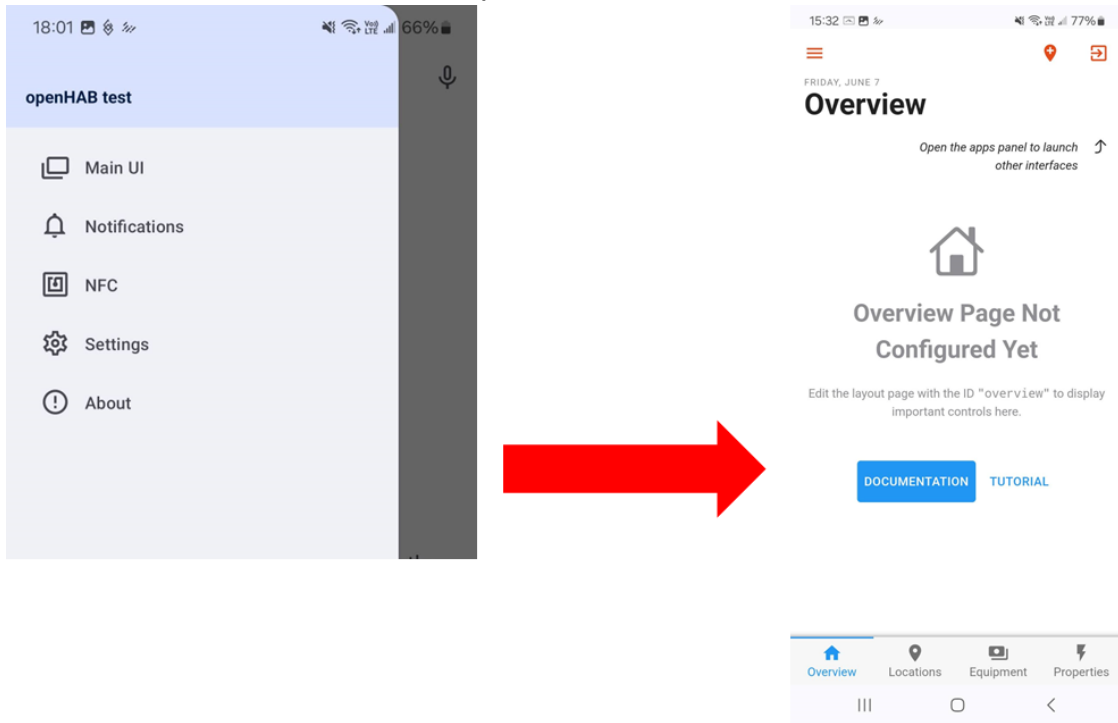
請參照此方式輸入: **http://10.33.122.185:8080/**


***注:** 紅字部分請勿直接照抄, 請複製您設定的 openHAB IP.(參閱 P.XX)

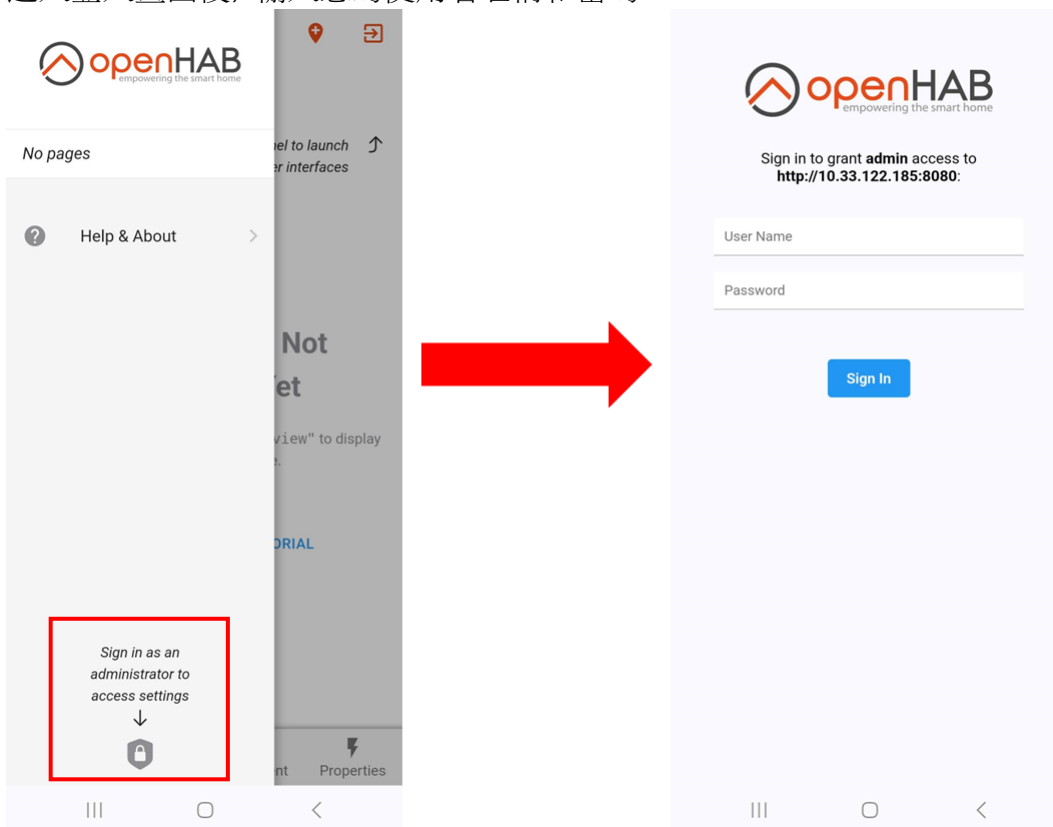
設定完後返回主畫面,看是否連線成功, 若是設定成功, 主畫面不會在顯示“ Your openHAB server is offline while the cloud instance is running ”.



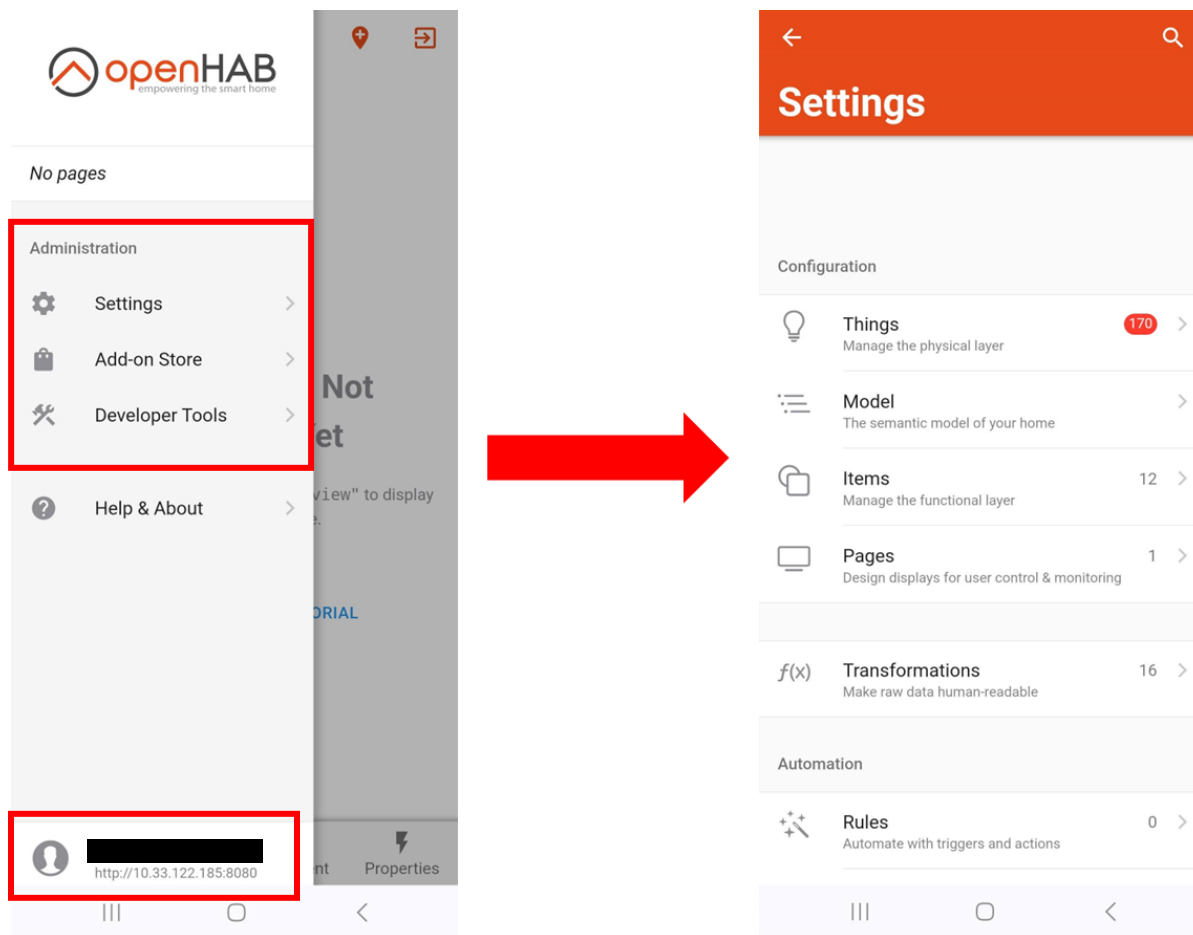
回到主畫面後請將畫面往左滑動，並點擊 **Main UI**，就會出現 **openHAB** 的系統介面(右圖)，但這並不代表你可以直接使用在 **openHAB** 的設定。



請在畫面點選左上角的 ，並在左下方點擊紅框中的 **Sign in**。進入登入畫面後，輸入您的使用者名稱和密碼。



成功登入後，即可使用您的個人設定。您也可隨時登出或切換別的使用者。

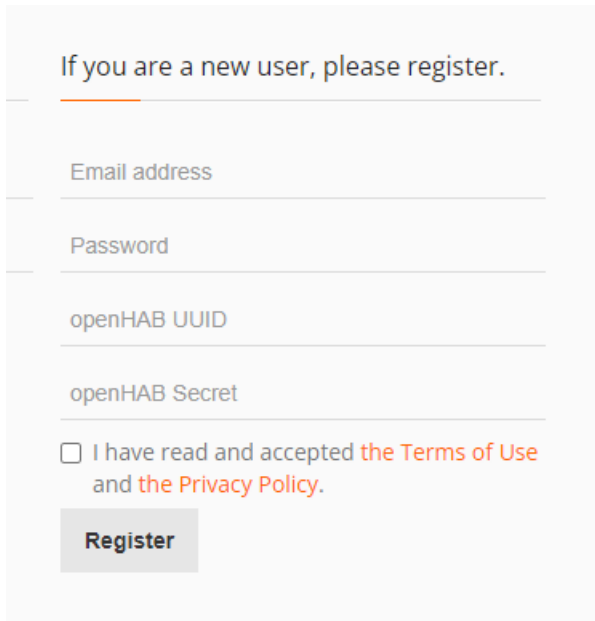


(6) 設定外網(Remote)教學:

- *注:** 1.請先在 openHAB 的 Add-on-store 裡面, 先安裝 openHAB Cloud Connector.
若沒有安裝此功能則無法進行遠端操控.
2.建議先完成內網設定後, 再來完成此設定.

(6-1)請先進入網站, 並註冊帳號.

註冊時會要求提供 openHAB 的 UUID 和 Secret



The image shows a registration form with the following fields and elements:

- Header: "If you are a new user, please register."
- Input fields: "Email address", "Password", "openHAB UUID", and "openHAB Secret".
- Checkbox: "I have read and accepted the Terms of Use and the Privacy Policy."
- Button: "Register"

要如何找到 UUID 和 Secret?

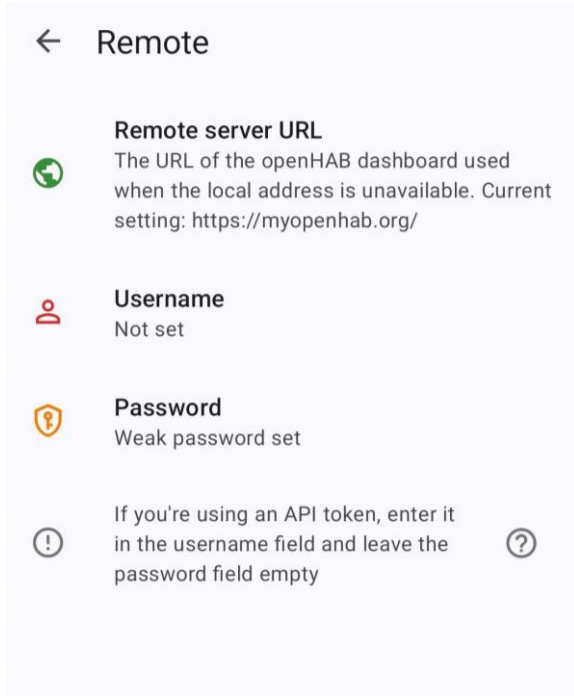
請打開您的 openHAB 資料夾, 並遵循下列指示:

1. **UUID:** 進入您的 openHAB 資料夾後, 點擊裡面的 userdata, 裡面的檔案中有一個 UUID 的檔案, 請用 notepad 開啟該檔案就可以知道您的 UUID.
2. **Secret:** 和 UUID 一樣, 進入到 userdata, 您會看到一個 openhabcloud 的檔案, 點進去後會看到一個名為 secret 的檔案, 透過 notepad 開啟該檔案就可以知道您的 secret.

註冊完畢後, 會有一封驗證信送到您的 E-mail, 確認收到後點擊驗證連結便正式完成帳戶註冊了.

現在我們回到 openHAB 的 APP, 請點選 settings 並進入 remote.

請在 Remote server URL 輸入 “<https://myopenhab.org>” (官方指定 domain)



使用者 和 密碼 就是您註冊 openHAB 的 username 和 password.

完成後請透過 行動網路 或 連結其他無線網路做測試, 若能成功連線, 代表著您可以在外面遠端控制 openHAB 裡面的設備.

6. 展示手冊圖片中 IP Power 的 code

6.1 9258 code

UID: http:url:bc4747fb52

label: IP9258 relay

thingTypeUID: http:url

configuration:

authMode: BASIC_PREEMPTIVE

ignoreSSLErrors: false

baseURL: http://10.33.122.48

password: "12345678"

delay: 0

stateMethod: GET

refresh: 30

commandMethod: GET

contentType: text/plain

timeout: 3000

bufferSize: 2048

username: admin

channels:

- id: id_IP9258_relay1

channelTypeUID: http:switch

label: IP9258 relay1

description: ""

configuration:

onValue: "1"

commandTransformation: MAP:9258_relay1.map

offValue: "0"

stateExtension: set.cmd?cmd=getpower

commandExtension: set.cmd?cmd=setpower+%2\$s

stateTransformation: REGEX:^\.*p61=(\d+).*

- id: id_IP9258_relay2

channelTypeUID: http:switch

label: IP9258 relay2

description: ""

configuration:

onValue: "1"

commandTransformation: MAP:9258_relay2.map

offValue: "0"

stateExtension: set.cmd?cmd=getpower

commandExtension: set.cmd?cmd=setpower+%2\$s

stateTransformation: REGEX:^\.*p62=(\d+).*

- id: id_IP9258_relay3

channelTypeUID: http:switch

label: IP9258 relay3

```

description: ""
configuration:
  onValue: "1"
  commandTransformation: MAP:9258_relay3.map
  offValue: "0"
  stateExtension: set.cmd?cmd=getpower
  commandExtension: set.cmd?cmd=setpower+%2$s
  stateTransformation: REGEX:.*p63=(\d+).*
```

- id: id_IP9258_relay4

```

channelTypeUID: http:switch
label: IP9258 relay4
description: ""
configuration:
  onValue: "1"
  commandTransformation: MAP:9258_relay4.map
  offValue: "0"
  stateExtension: set.cmd?cmd=getpower
  commandExtension: set.cmd?cmd=setpower+%2$s
  stateTransformation: REGEX:.*p64=(\d+).*
```

6.2 98 系列 code

UID: http:url:9b4e47dc4d

label: http_98XX_relay

thingTypeUID: http:url

configuration:

```

authMode: BASIC
ignoreSSLErrors: false
baseUrl: http://10.33.122.52
password: "12345678"
delay: 0
stateMethod: GET
refresh: 30
commandMethod: POST
contentType: application/json
timeout: 3000
bufferSize: 2048
username: admin
```

channels:

```

- id: http_9850_relay1
  channelTypeUID: http:switch
  label: http 9850 relay1
  description: ""
  configuration:
    onValue: "1"
    commandTransformation: MAP:http_cmd1.map
```

```

    offValue: "0"
    stateExtension: json.cmd?getpower
    commandExtension: json.cmd?
    stateTransformation: JSONPATH:$.result.RL[0].state
- id: http_9850_relay2
  channelTypeUID: http:switch
  label: http 9850 relay2
  description: ""
  configuration:
    onValue: "1"
    commandTransformation: MAP:http_cmd1.map
    offValue: "0"
    stateExtension: json.cmd?getpower
    commandExtension: json.cmd?
    stateTransformation: JSONPATH:$.result.RL[1].state
- id: http_9850_relay3
  channelTypeUID: http:switch
  label: http 9850 relay3
  description: ""
  configuration:
    onValue: "1"
    commandTransformation: MAP:http_cmd1.map
    offValue: "0"
    stateExtension: json.cmd?getpower
    commandExtension: json.cmd?
    stateTransformation: JSONPATH:$.result.RL[2].state
- id: http_9850_relay4
  channelTypeUID: http:switch
  label: http 9850 relay4
  description: ""
  configuration:
    onValue: "1"
    commandTransformation: MAP:http_cmd1.map
    offValue: "0"
    stateExtension: json.cmd?getpower
    commandExtension: json.cmd?
    stateTransformation: JSONPATH:$.result.RL[3].state

```

6.3 Generic MQTT thing code

UID: mqtt:topic:e202ba2650

label: Generic MQTT Thing

thingTypeUID: mqtt:topic

configuration:

 payloadNotAvailable: offline

 payloadAvailable: online

 transformationPattern: JS:mqtt_availability.js

 availabilityTopic: mac/0076XXXXXXXX/status

bridgeUID: mqtt:broker:8496c99244

channels:

- id: my98XX_mqtt_switch1

 channelTypeUID: mqtt:switch

 label: my9850 mqtt switch1

 description: ""

 configuration:

 qos: 0

 transformationPatternOut: MAP:mqtt_cmd.map

 commandTopic: mac/0076XXXXXXXX/cmd

 stateTopic: mac/0076XXXXXXXX/cmd_ack/id_my_openhab

 transformationPattern: JINJA:"{% if (value_json.cmd == 'getpower' and value_json.result.RL[0].state == 1) or (value_json.tag == 'r1_on' and value_json.result.message == 'success') %} 'p1_1' {% elif (value_json.cmd == 'getpower' and value_json.result.RL[0].state == 0) or (value_json.tag == 'r1_off' and value_json.result.message == 'success') %} 'p1_0' {% else %} 'p1_0' {% endif %}"

 off: p1_0

 on: p1_1

- id: my98XX_mqtt_switch2

 channelTypeUID: mqtt:switch

 label: my98XX mqtt switch2

 description: ""

 configuration:

 qos: 0

 transformationPatternOut: MAP:mqtt_cmd.map

 commandTopic: mac/0076XXXXXXXX/cmd

 stateTopic: mac/0076XXXXXXXX/cmd_ack/id_my_openhab

 transformationPattern: JINJA:"{% if (value_json.cmd == 'getpower' and value_json.result.RL[1].state == 1) or (value_json.tag == 'r2_on' and value_json.result.message == 'success') %} 'p2_1' {% elif (value_json.cmd == 'getpower' and value_json.result.RL[1].state == 0) or (value_json.tag == 'r2_off' and value_json.result.message == 'success') %} 'p2_0' {% else %} 'p2_0' {% endif %}"

```

    off: p1_0
    on: p1_1
- id: my98XX_mqtt_switch3
  channelTypeUID: mqtt:switch
  label: my98XX mqtt switch3
  description: ""
  configuration:
    qos: 0
    transformationPatternOut: MAP:mqtt_cmd.map
    commandTopic: mac/0076XXXXXXXXX /cmd
    stateTopic: mac/0076XXXXXXXXX /cmd_ack/id_my_openhab
    transformationPattern: JINJA:"{% if (value_json.cmd == 'getpower' and
      value_json.result.RL[1].state == 1) or (value_json.tag == 'r3_on' and
      value_json.result.message == 'success') %} 'p3_1' {% elif
      (value_json.cmd == 'getpower' and value_json.result.RL[1].state == 0) or
      (value_json.tag == 'r3_off' and value_json.result.message ==
      'success')%} 'p3_0' {% else %} 'p3_0' {% endif %}"
    off: p1_0
    on: p1_1
- id: my98XX_mqtt_switch4
  channelTypeUID: mqtt:switch
  label: my98XX mqtt switch4
  description: ""
  configuration:
    qos: 0
    transformationPatternOut: MAP:mqtt_cmd.map
    commandTopic: mac/0076XXXXXXXXX /cmd
    stateTopic: mac/0076XXXXXXXXX /cmd_ack/id_my_openhab
    transformationPattern: JINJA:"{% if (value_json.cmd == 'getpower' and
      value_json.result.RL[1].state == 1) or (value_json.tag == 'r4_on' and
      value_json.result.message == 'success') %} 'p4_1' {% elif
      (value_json.cmd == 'getpower' and value_json.result.RL[1].state == 0) or
      (value_json.tag == 'r4_off' and value_json.result.message ==
      'success')%} 'p4_0' {% else %} 'p4_0' {% endif %}"
    off: p1_0
    on: p1_1

```

6.4 屬性設定: (如: 電流, 電壓, 溫度)

在 P.33 的 State Transformation 中輸入:

***注意:** 每個 relay 的屬性不一定一致. 請勿直接複製一樣的 code 做設置.

(1)溫度: JINJA:{{value_json.result.global_measure.temperature}}

電壓: JINJA:{{value_json.result.global_measure.voltage}}

(2)電流: JINJA:{{value_json.result.RL[0].current}}

免責聲明:

該手冊的內容僅供 **IP Power** 使用者參考和一般訊息目的，並且可能隨時更改而不事先通知。**Aviosys** 睿意科技對於本手冊上提供 **openHAB** 應用在 **IP Power** 訊息的準確性、完整性和可靠性不作任何明示或暗示的擔保或聲明。

任何在該手冊上獲取的資訊都是基於用戶自己的風險。我們不承擔任何直接或間接的損失或損害，包括但不限於利潤損失、商業中斷、數據損失或其他財務損失，即使我們已被告知可能發生此類損害。

該手冊可能包含第三方網站的鏈接或引用。我們對於這些第三方網站的內容或可用性不承擔任何責任，並且不對其進行背書、擔保或聲明。您應自行判斷和承擔使用這些第三方網站的風險。

我們保留隨時更改或修改本免責聲明的權利。建議您定期查看本頁面以了解任何更改。通過繼續使用該手冊，即表示您同意遵守這些條款和條件。

如果您對於本免責聲明有任何問題，請與我們聯繫。