

Domótica

openHAB + The Things Network + Alexa

Juan Félix Mateos Barrado
jfmateos@educa.madrid.org
juanfelixmateos@gmail.com

Medialab Prado
Noviembre 2019

Objetivo

Integrar un nodo de The Things Network en openHAB para poder conocer su temperatura y controlar su LED utilizando comandos de voz a través de Alexa.

openHAB

- Open Home Automation BUS
- Creado por **Kai Kreuzer** (2010) en Java
- Software abierto de domótica patrocinado por KNX, Z-Wave...
- Funciona en Windows, Linux, MacOS, Raspberry Pi...
- A partir de openHAB se creó Eclipse SmartHome (2013) para aplicaciones comerciales.

Conceptos básicos

- **Things, Items y Channels**

- Un sensor de temperatura, presión y humedad es una thing (cosa física), que ofrece 3 canales (temperatura, presión y humedad), con los que el usuario podrá interactuar a través de un item (elemento virtual)

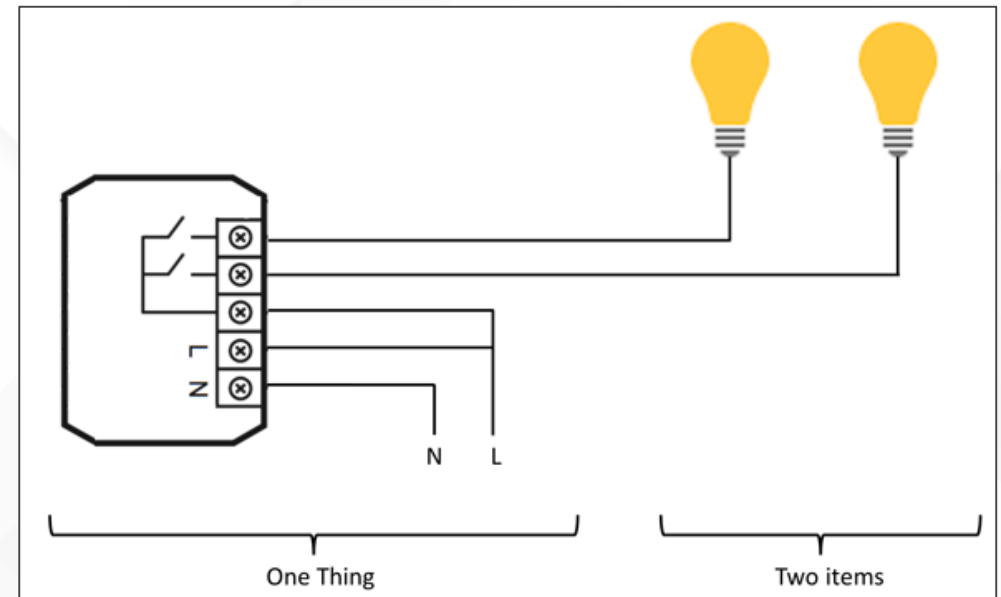
- **Bindings**

- Son las extensiones de openHAB que nos permiten usar distintos tipos de things (Z-Wave, Zigbee...)

- **Rules**

- **Sitemaps**

- **Dashboards**



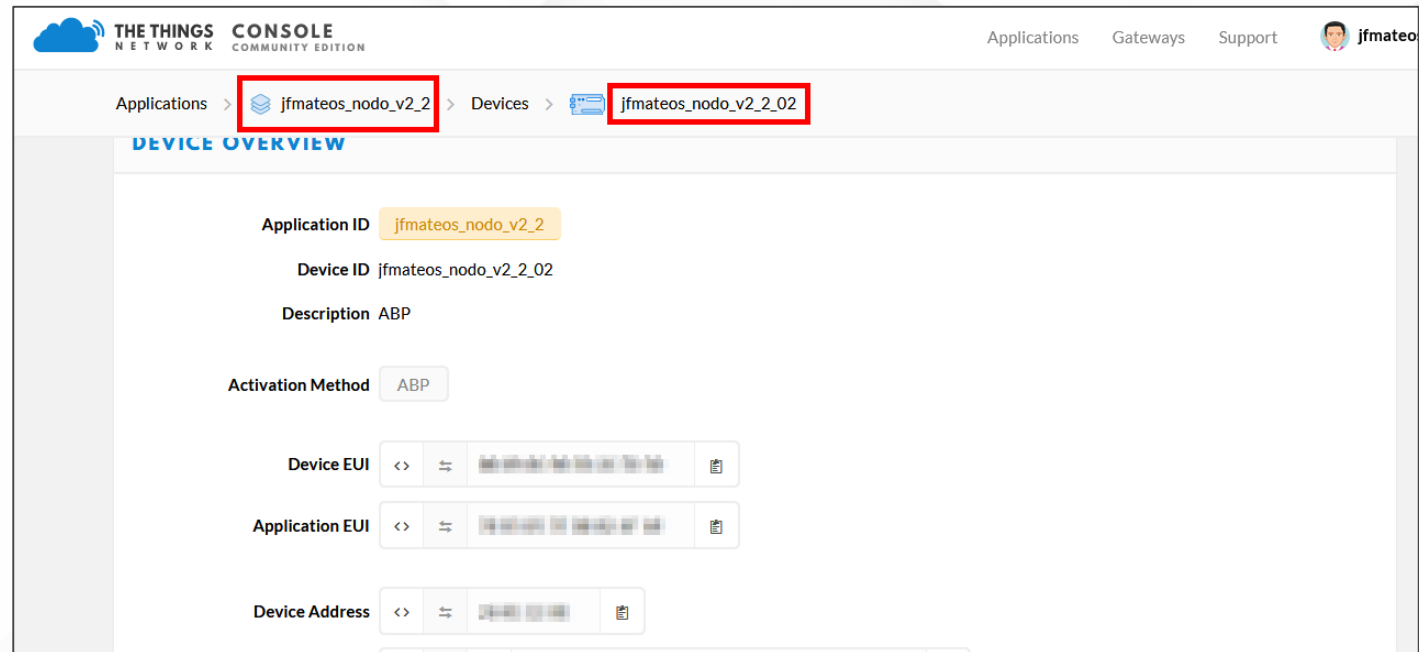
Orden del día

- Crear una aplicación y un dispositivo en The Things Network
- Instalar Java
- Instalar openHAB
- Instalar el binding JSONPath
- Crear una thing de tipo Generic MQTT Thing con 2 canales vinculados a 2 items:
 - Temperatura
 - LED
- Crear una regla que active/desactive el LED en el umbral de temperatura de 25º
- Crear un sitemap que muestre la fecha/hora, y los controles de Temperatura y el LED.
- Instalar el binding openHAB Cloud Connector
- Instalar la app de openHAB para el móvil.
- Instalar el binding Rest documentation
- Añadir metadatos a los items para poder controlarlos con Alexa.
- Instalar Alexa en el móvil.
- Configurar el skill de openHAB

Crear aplicación y dispositivo en TTN

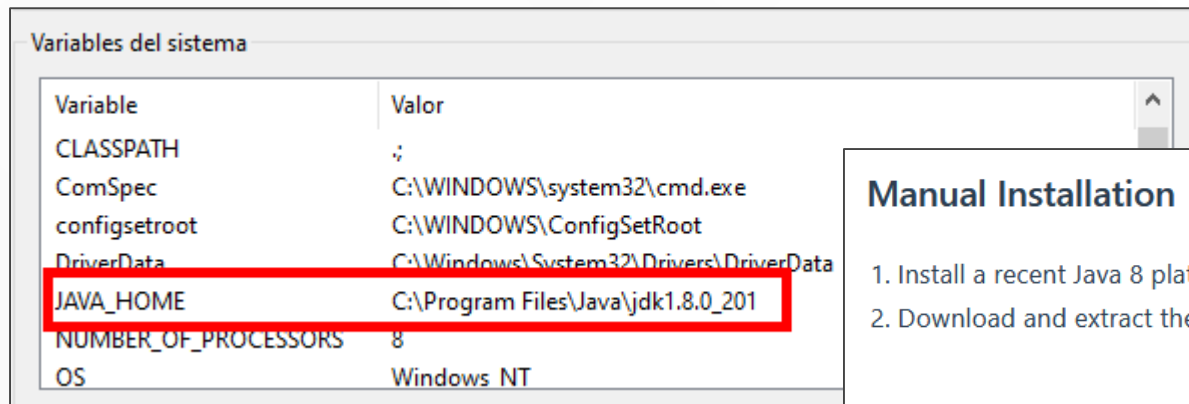
- No es imprescindible un nodo físico. Podemos simularlo con la siguiente carga de pago
 - 01 02 01 2D 02 00 01 **03 67 00 E2** 04 73 24 77 05 68 7D 06 01 01 (T=22.6)
 - 01 02 01 2D 02 00 01 **03 67 01 07** 04 73 24 77 05 68 7D 06 01 01 (T=26.3)

```
{  
  "analog_in_1": 3.01,  
  "barometric_pressure_4": 933.5,  
  "digital_in_2": 1,  
  "digital_out_6": 1,  
  "relative_humidity_5": 62.5,  
  "temperature_3": 22.6  
}
```



Instalación en Windows 1/2

- Requiere Java 8 (se recomienda la versión Zulu)
- Asegurarse de que está definida la variable de entorno JAVA_HOME



Manual Installation

1. Install a recent Java 8 platform (we recommend [Zulu](#)), see [prerequisites](#)
2. Download and extract the openHAB runtime distribution from <https://bintray.com/openhab/mvn/openhab-distro>:

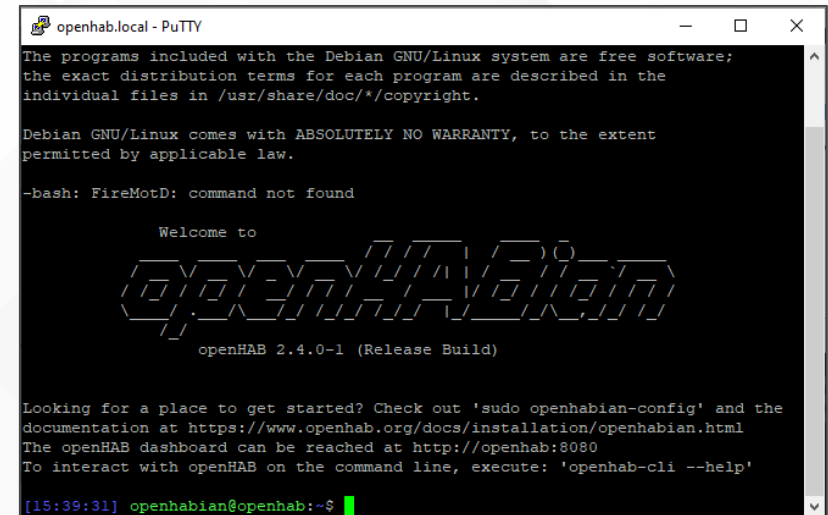
[Download openHAB 2.4.0 Stable Runtime](#)

Instalación en Windows 2/2

- Descargar el archivo comprimido de openHAB
- Descomprimirlo en una carpeta de nuestro disco duro; por ejemplo C:/openHAB
- Ejecutar el archivo start.bat para iniciar openHAB
- Acceder con el navegador web a localhost:8080

Iniciar y detener openHAB

- A través de la consola:
 - Windows: C:\openHAB\runtime\bin\karaf.bat
 - logout
 - Raspberry pi: Putty/SSH
 - Usuario: openHABian
 - Contraseña: openHABian
 - sudo shutdown -h now



```
openhab.local - PuTTY
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

-bash: FireMotD: command not found

Welcome to

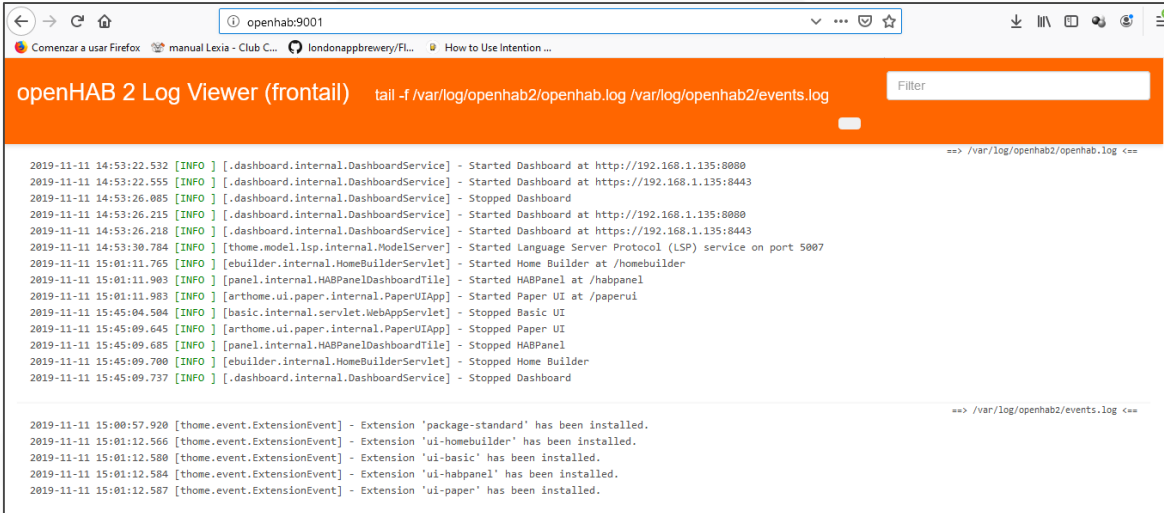
  openHAB
  openHAB 2.4.0-1 (Release Build)

Looking for a place to get started? Check out 'sudo openhabian-config' and the
documentation at https://www.openhab.org/docs/installation/openhabian.html
The openHAB dashboard can be reached at http://openhab:8080
To interact with openHAB on the command line, execute: 'openhab-cli --help'

[15:39:31] openhabian@openhab:~$
```

Logs

- Hay 2 logs importantes:
 - events.log → Todo lo relacionado con las things
 - openHAB.log → Todo lo relacionado con los bindings
- Windows
 - C:\openHAB\userdata\logs
- Raspberry pi
 - <http://openHAB:9001/>



The screenshot shows the 'openHAB 2 Log Viewer (frontail)' interface in a web browser. The address bar shows 'openhab:9001'. The page title is 'openHAB 2 Log Viewer (frontail)' and the URL bar shows 'tail -f /var/log/openhab2/openhab.log /var/log/openhab2/events.log'. There is a 'Filter' input field. The log content is displayed in a monospace font, showing various system events and service status changes. The logs are organized into two sections, each with a header indicating the file being viewed: '/var/log/openhab2/openhab.log' and '/var/log/openhab2/events.log'.

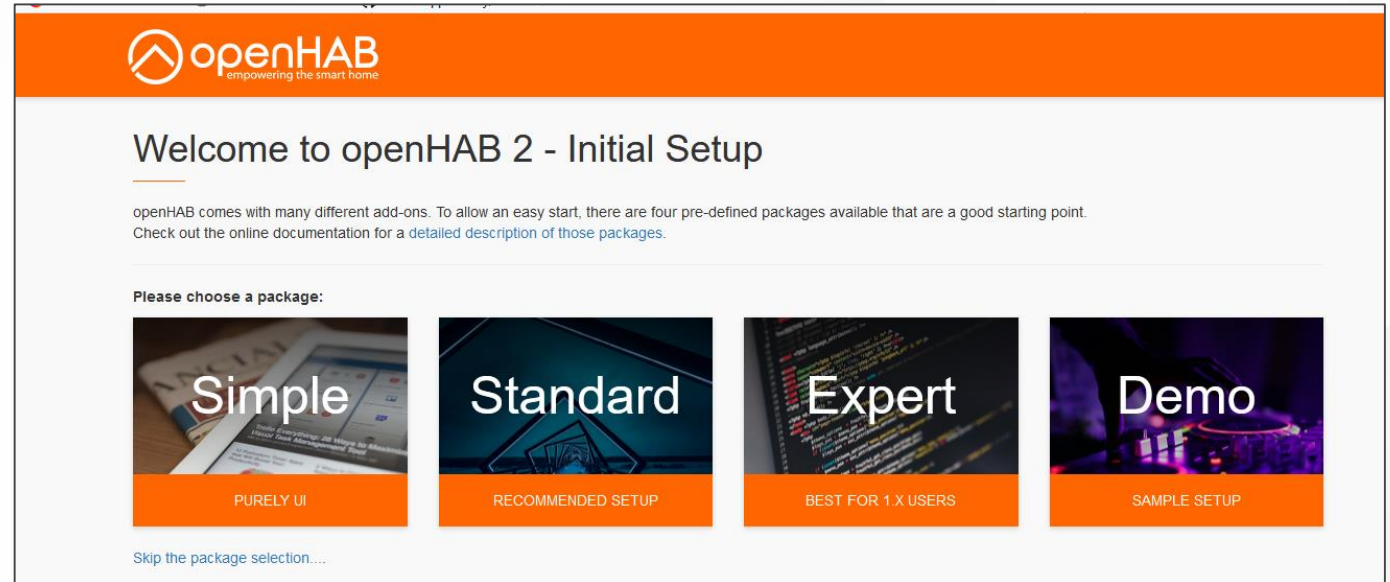
```
2019-11-11 14:53:22.532 [INFO] [.dashboard.internal.DashboardService] - Started Dashboard at http://192.168.1.135:8080
2019-11-11 14:53:22.555 [INFO] [.dashboard.internal.DashboardService] - Started Dashboard at https://192.168.1.135:8443
2019-11-11 14:53:26.085 [INFO] [.dashboard.internal.DashboardService] - Stopped Dashboard
2019-11-11 14:53:26.215 [INFO] [.dashboard.internal.DashboardService] - Started Dashboard at http://192.168.1.135:8080
2019-11-11 14:53:26.218 [INFO] [.dashboard.internal.DashboardService] - Started Dashboard at https://192.168.1.135:8443
2019-11-11 14:53:30.784 [INFO] [thome.model.lsp.internal.ModelServer] - Started Language Server Protocol (LSP) service on port 5007
2019-11-11 15:01:11.765 [INFO] [ebuilder.internal.HomeBuilderServlet] - Started Home Builder at /homebuilder
2019-11-11 15:01:11.983 [INFO] [panel.internal.HABPanelDashboardTile] - Started HABPanel at /habpanel
2019-11-11 15:01:11.983 [INFO] [arhome.ui.paper.internal.PaperUIApp] - Started Paper UI at /paperui
2019-11-11 15:45:04.584 [INFO] [basic.internal.servlet.WebAppServlet] - Stopped Basic UI
2019-11-11 15:45:09.645 [INFO] [arhome.ui.paper.internal.PaperUIApp] - Stopped Paper UI
2019-11-11 15:45:09.685 [INFO] [panel.internal.HABPanelDashboardTile] - Stopped HABPanel
2019-11-11 15:45:09.700 [INFO] [ebuilder.internal.HomeBuilderServlet] - Stopped Home Builder
2019-11-11 15:45:09.737 [INFO] [.dashboard.internal.DashboardService] - Stopped Dashboard

2019-11-11 15:00:57.920 [thome.event.ExtensionEvent] - Extension 'package-standard' has been installed.
2019-11-11 15:01:12.566 [thome.event.ExtensionEvent] - Extension 'ui-homebuilder' has been installed.
2019-11-11 15:01:12.580 [thome.event.ExtensionEvent] - Extension 'ui-basic' has been installed.
2019-11-11 15:01:12.584 [thome.event.ExtensionEvent] - Extension 'ui-habpanel' has been installed.
2019-11-11 15:01:12.587 [thome.event.ExtensionEvent] - Extension 'ui-paper' has been installed.
```


Configuración inicial

Se recomienda utilizar la opción Standard, que instalará las interfaces:


- Paper UI
- Basic UI
- Hab PANEL
- Home Builder



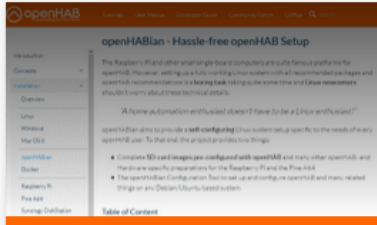
Interfaces (1/2)



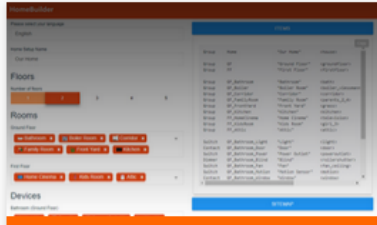
empowering the smart home



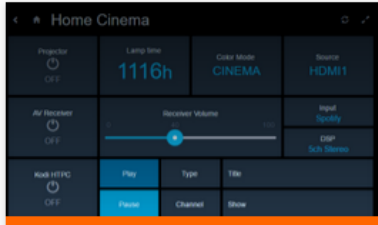
OPENHAB LOG VIEWER



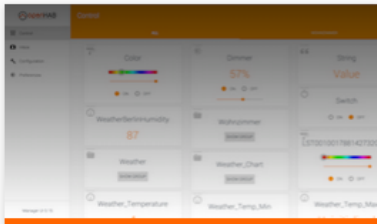
OPENHABIAN HELP



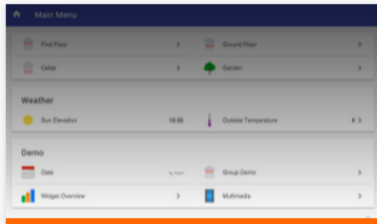
HOME BUILDER



HABPANEL



PAPER UI



BASIC UI

Getting started? Please refer to the [online documentation](#).

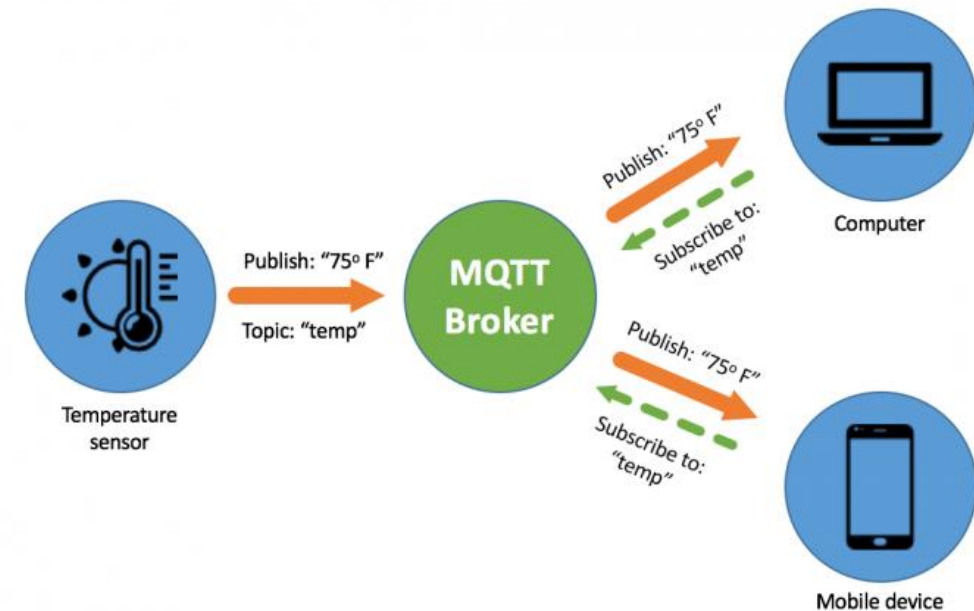
openHAB 2.4.0 Release Build

Interfaces (2/2)

- **Paper UI:** Ideal para configurar openHAB.
- **Basic UI:** Ideal para mostrar los Site maps
- **HABPanel:** Sirve para diseñar Dashboards.
- **Home Builder:** Sirve para generar una configuración inicial compuesta por:
 - Items
 - Sitemaps
 - Dashboards

MQTT

- Broker
- Suscripción/Publicación
 - Topic
- Vamos a vincular openHAB con el bróker MQTT de The Things Network



MQTT

- Instalar el binding MQTT
- Crear un thing de tipo MQTT Broker



MQTT: Configurar la thing MQTT Broker

- Broker: eu.thethings.network
- Username: Nombre de la aplicación en TTN
- Password: Default key de TTN

Configuration Parameters

Configure parameters for the thing.

Broker Hostname/IP

eu.thethings.network

The IP/Hostname of the MQTT broker

SHOW LESS

Broker Port

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

Client ID

paho41748559395100

Use a fixed client ID. Defaults to empty which means a user ID is generated for this connection.

Secure connection

Uses TLS/SSL to establish a connection

Quality of Service

At most once (0)

Retain messages

Retained messages are stored on the MQTT broker and other clients can retrieve the value at any time

Reconnect time

60000

Reconnect time in ms. If a connection is lost, the binding will wait this time before it tries to reconnect.

Last will message

The last will message.

Last will QoS

At most once (0)

The quality of service parameter of the last will.

Password

.....

The MQTT password

Certificate hash

If **certificatepin** is set this hash is used to verify the connection. Clear to allow a new certificate pinning on the next connection attempt. If empty will be filled automatically by the next successful connection. An example input would be: [...more](#)

Heartbeat

60000

Keep alive / heartbeat timer in ms. It can take up to this time to determine if a server connection is lost. A lower value may keep the broker unnecessarily busy for no or little additional value.

Last will topic

Defaults to empty and therefore disables the last will.

Username

jfmateos_molinera

The MQTT username

Certificate pinning

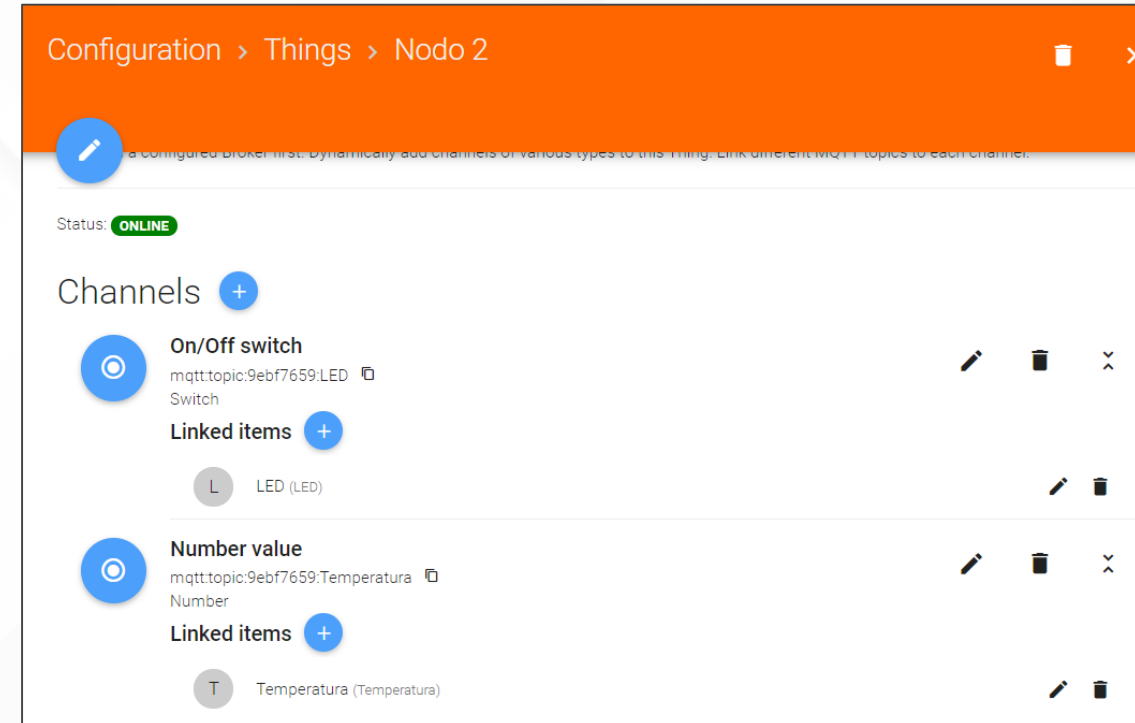
If this and SSL is set: After the next connection has been successfully established, the certificate is pinned. The connection will be refused if another certificate is used. Clear **certificate** to allow a new certificate for the next connection attempt: [...more](#)

Public key pinning

If this and SSL is set: After the next connection has been successfully established, the public key of the broker is pinned. The connection will be refused if another public key is used. Clear **publickey** to allow a new public key for the next connection attempt: [...more](#)

MQTT: Crear una Generic MQTT Thing

- El thing Broker es el soporte general para poder usar MQTT en openHAB, pero adicionalmente tenemos que crear un thing de tipo Generic MQTT Thing para cada elemento que queramos controlar.
- Crear una thing de tipo Generic MQTT Thing con 2 canales vinculados a los siguientes items:
 - Number value: **Temperatura**
 - On/Off Switch: **LED**



MQTT: Canal para recibir uplink

- Configurar el canal de tipo Number Value del siguiente modo

Configure channel

MQTT state topic

[+/devices/+/up/temperature_3](#)

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

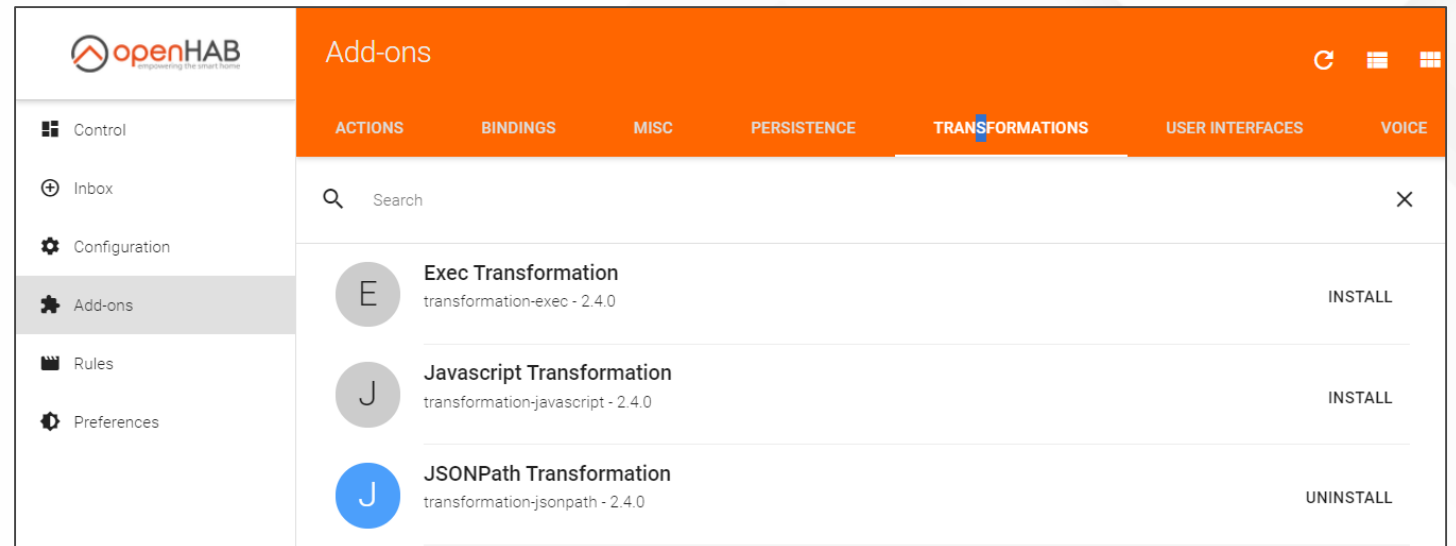
[SHOW MORE](#)

[CANCEL](#) [SAVE](#)

El 20 de noviembre de 2019 han desactivado la publicación de los decoded fields en topics MQTT independientes; consecuentemente este sistema no funciona (ver diapositiva siguiente)

MQTT: Canal para recibir uplink (alternativa)

- En lugar de suscribirnos al topic concreto de la temperatura, nos suscribiremos al topic completo, y extraeremos de él la temperatura mediante una transformación.
 - Instalar el binding JSONPath Transformation



MQTT: Canal para recibir uplink (alternativa)

- Configurar el canal de temperatura del siguiente modo

Configure channel

MQTT state topic

+/devices/+/up

An MQTT topic that this thing will subscribe to, to receive the state. This can be left empty, the channel will be state-less then and will publish values non-retained.

MQTT command topic

An MQTT topic that this thing will send a command to. This can be left empty

Absolute minimum

This configuration represents the minimum of the allowed range. For a percentage channel that equals zero percent.

Absolute maximum

This configuration represents the maximum of the allowed range. For a percentage channel that equals one-hundred percent.

SHOW LESS

Incoming value transformation

JSONPATH:\$payload_fields.temperature_3

Applies a transformation 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 }}}. Any supported transform: [more](#)

Outgoing value format

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: [more](#)

CANCEL

SAVE

MQTT: Canal para enviar downlink

- Configurar el canal de tipo On/Off Switch del siguiente modo
 - **BgBk/w==** y **BgAA/w==** son la codificación en BASE64 de 06 00 64 FF y 06 00 00 FF, que son la carga de pago que espera recibir el nodo para encender y apagar el led, respectivamente
 - jfmateos_nodo_v2_2/devices/jfmateos_nodo_v2_2_02/down
 - {"port":99,"confirmed":true,"payload_raw":"%s"}

Configure channel

MQTT state topic

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.

SHOW LESS

Incoming value transformation

Applies a transformation 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 }}}. Any supported transformation service can be used.

☐ 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.

On/open value

BgBk/w==

A number (like 1, 10) or a string (like "enabled") that is recognised as on/open state. You can

MQTT command topic

jfmateos_nodo_v2_2/devices/jfmateos_nodo_v2_2_02/down

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

Outgoing value format

{ "port":99,"confirmed":true,"payload_raw":"%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 can use "%.4f".

☐ Is command

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

Off/closed value

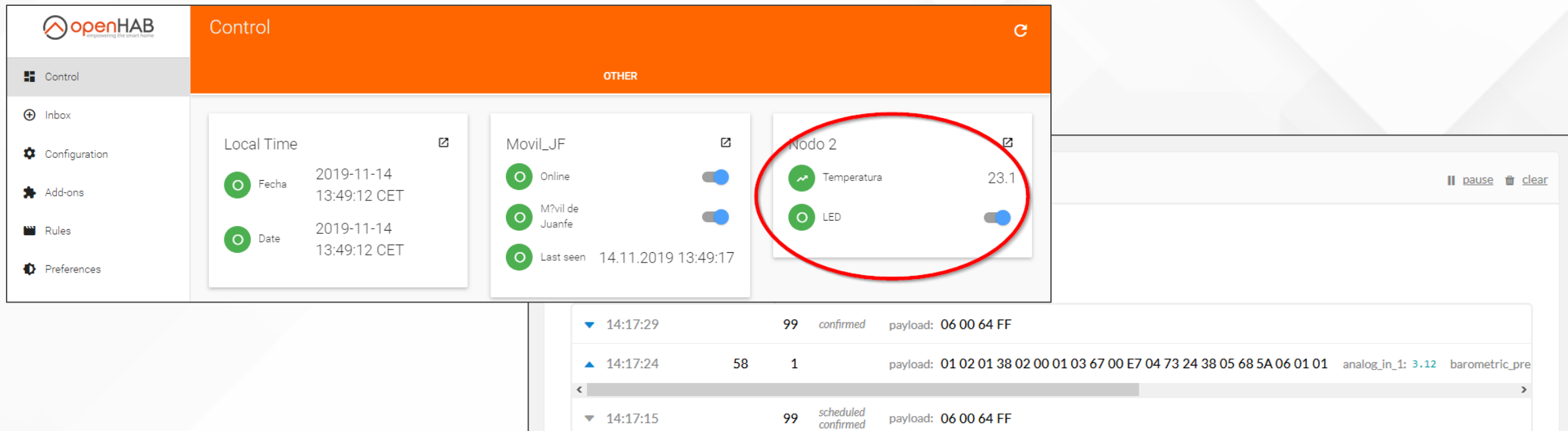
BgAA/w==

A number (like 0, -10) or a string (like "disabled") that is recognised as off/closed state. You can

CANCEL SAVE

MQTT: Canal para enviar downlink

- En la versión 2.4 de openHAB hay un bug que impide que se inicie automáticamente el cliente MQTT
 - Para resolverlo hay que cerrar openHAB (logout) y volver a iniciarlo.



The screenshot displays the openHAB Control interface. The left sidebar contains navigation links: Control, Inbox, Configuration, Add-ons, Rules, and Preferences. The main area is titled 'Control' and features three widgets: 'Local Time' showing the current date and time, 'Movil_JF' showing online status and last seen time, and 'Nodo 2' showing temperature and LED status. A red circle highlights the 'Nodo 2' widget. At the bottom, a MQTT message log is visible, showing a list of messages with timestamps, QoS, and payloads.

Timestamp	QoS	Message Type	Payload	Additional Info
14:17:29	99	confirmed	06 00 64 FF	
14:17:24	58	1	01 02 01 38 02 00 01 03 67 00 E7 04 73 24 38 05 68 5A 06 01 01	analog_in_1: 3.12 barometric_pre
14:17:15	99	scheduled confirmed	06 00 64 FF	

Reglas

Nos permiten reaccionar ante:

- el cambio de estado de los **ítems**
 - Things y canales (de forma más limitada)
- instantes (fecha y/u hora) concretos
- eventos del sistema (que se inicie o se apague)

con acciones como:

- enviar un comando a un item
- utilizar un servicio (enviar un mail, notificación...)
- La sintaxis está basada en Xbase

Reglas: Encender/apagar el LED en el umbral de 25º

Crear en la carpeta C:\openHAB\conf\rules un archivo llamado default.rules con el siguiente contenido:

```
rule "LED"
when
    Item Temperatura changed
then
    if (Temperatura.state >= 25 && previousState<25) {
        LED.sendCommand(ON)
    } else {
        if(Temperatura.state < 25 && previousState>=25) {
            LED.sendCommand(OFF)
        }
    }
end
```


Sitemaps

Son representaciones destinadas al usuario para conocer el estado del sistema o interactuar con él.

Están compuestos por elementos de distintos tipos:

- Pulsadores
- Deslizadores
- Selectores de color
- ...

Los sitemaps se crean en archivos de texto con la extensión **sitemap** dentro de la carpeta:

- C:\openHAB\conf\sitemaps

En Paper UI disponemos de la sección Control, pero sólo sirve para Things (los sitemaps y dashboards funcionan principalmente a nivel de items).

Sitemaps

Cada sitemap debe empezar con un elemento de tipo sitemap

```
sitemap <nombre> label="<título de la pantalla inicial>" {  
    [resto de elementos]  
}
```

El **<nombre>** debe coincidir con el nombre del archivo; por ejemplo, si el sitemap se llama habitacion_invitados, el archivo deberá llamarse habitacion_invitados.sitemap.

Sitemaps

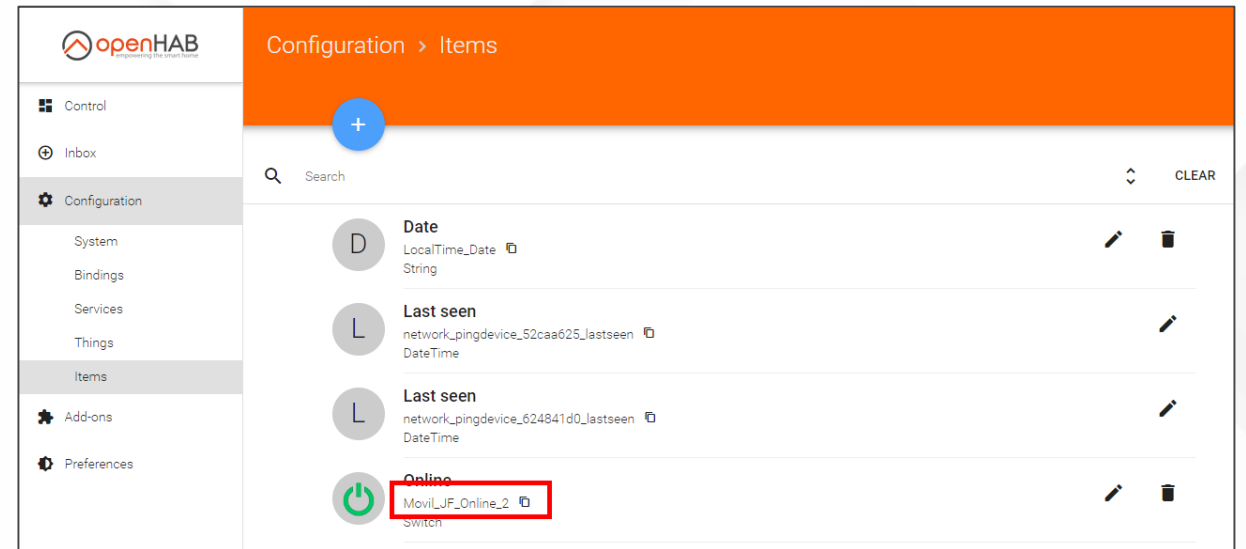
Cada elemento tendrá la siguiente sintaxis:

tipo item=nombre label="Rotulo"

Donde tipo puede ser:

- Switch
- Slider
- Text
- Image
- Frame (para agrupar)
- ...

Y nombre es el nombre del item.

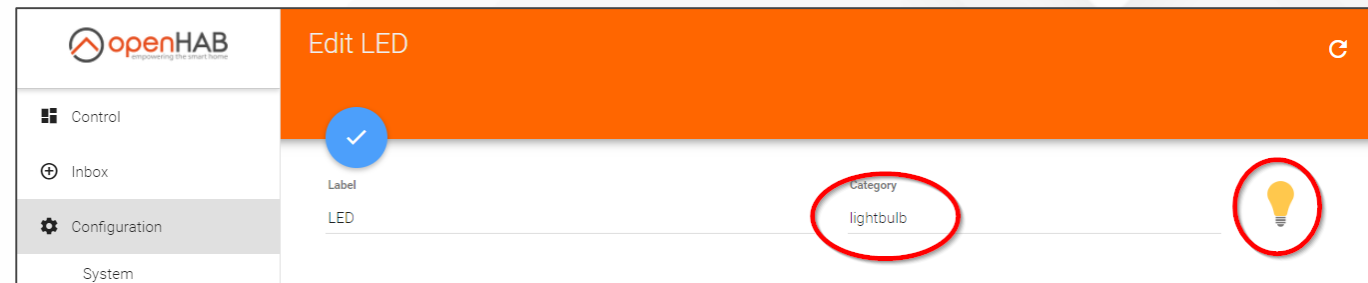
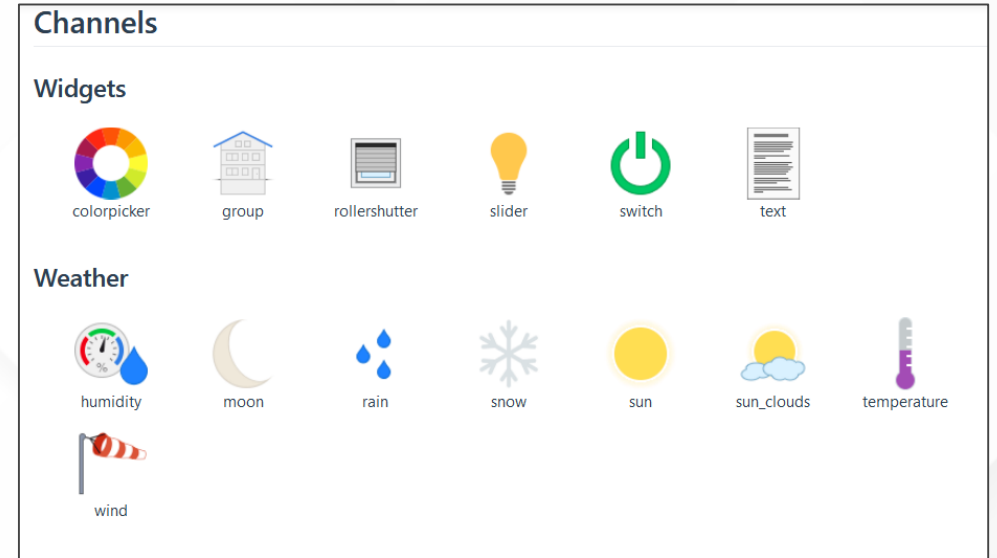


Sitemaps

Si lo deseamos podemos cambiar el icono del **item** en el sitemap eligiendo cualquiera de los disponibles en:

<https://www.openHAB.org/docs/configuration/iconsets/classic/>

Y escribiendo su nombre en **Category**



Sitemaps: things

Crear el archivo C:\openHAB\conf\sitemaps**ttn.sitemap** con el siguiente contenido:

```
sitemap ttn label="<The Things Network Madrid>" {  
    Switch item=LED label="LED"  
    Text item=Temperatura label="Salon [%.1f °C]" icon="temperature"  
}
```

Sitemaps: Basic UI

Configura Basic UI para que utilice como sitemap por defecto el que hemos creado anteriormente (habitación_invitados).

The image shows the openHAB configuration interface. On the left, a sidebar menu has 'Services' highlighted with a red box and a red arrow labeled '1'. The main area is titled 'Configuration > Services' and has tabs for MQTT, IO, MISC, and UI. The 'UI' tab is selected and highlighted with a red box and a red arrow labeled '2'. Below the tabs, there are two service cards: 'Basic UI' (org.eclipse.smarthome.basicui) and 'Panel' (openhab.habpanel). The 'Basic UI' card has a 'CONFIGURE' button highlighted with a red box and a red arrow labeled '3'. To the right, a 'Configure Basic UI' dialog box is open, showing various settings. The 'Default Sitemap' setting is highlighted with a red box, and its value 'ttn' is also highlighted with a red box.

Configuration > Services

MQTT IO MISC UI

Basic UI
org.eclipse.smarthome.basicui
CONFIGURE

Panel
openhab.habpanel
CONFIGURE

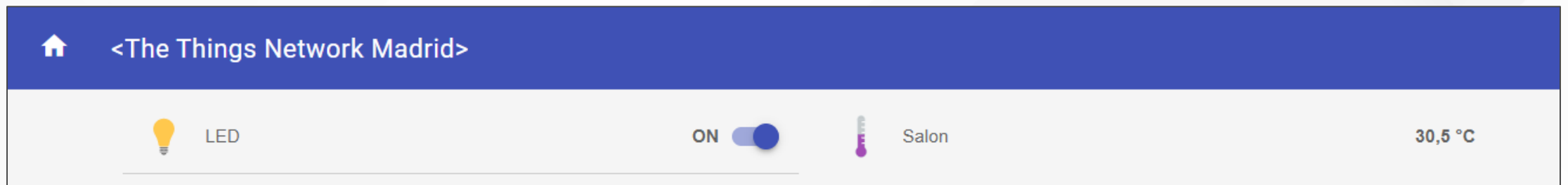
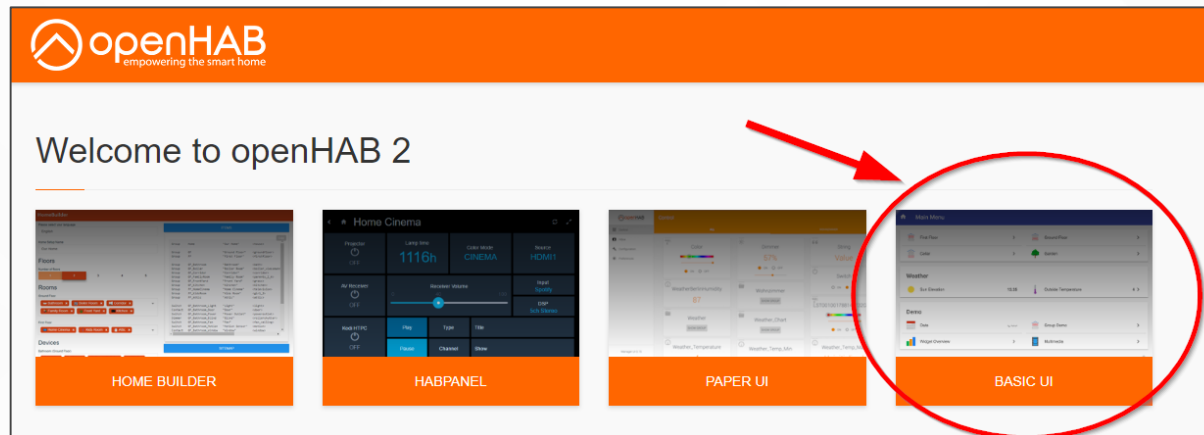
Configure Basic UI org.eclipse.smarthome.basicui

Theme: Default
Icon Format: Vector
Enable Icons: Enable
Condensed Layout: Disable
Capitalize Values: Disable
Default Sitemap: ttn

EXPERT MODE CANCEL SAVE

Sitemaps: Basic UI

Acceder a Basic UI a través de <http://localhost:8080>



Sitemaps

Además de crear ítems a través de Paper UI, también podemos hacerlo a través de documentos de texto con la extensión .items en la carpeta

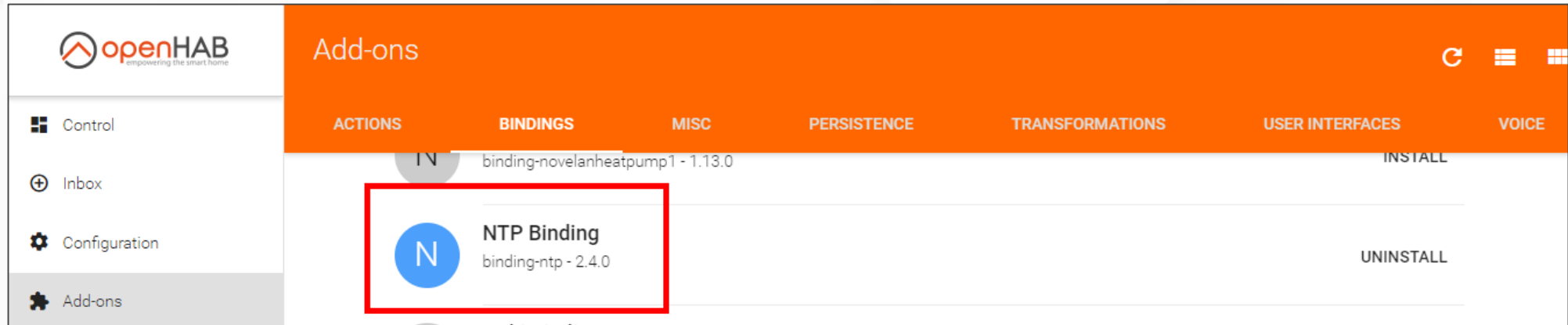
`C:\openHAB\conf\items`

Por ejemplo, vamos a incluir en el sitemap un item que muestre la fecha:

1. Importar el binding NTP.
2. Crear la thing correspondiente a través de Inbox.
3. Configurar el item a través de un archivo .items.

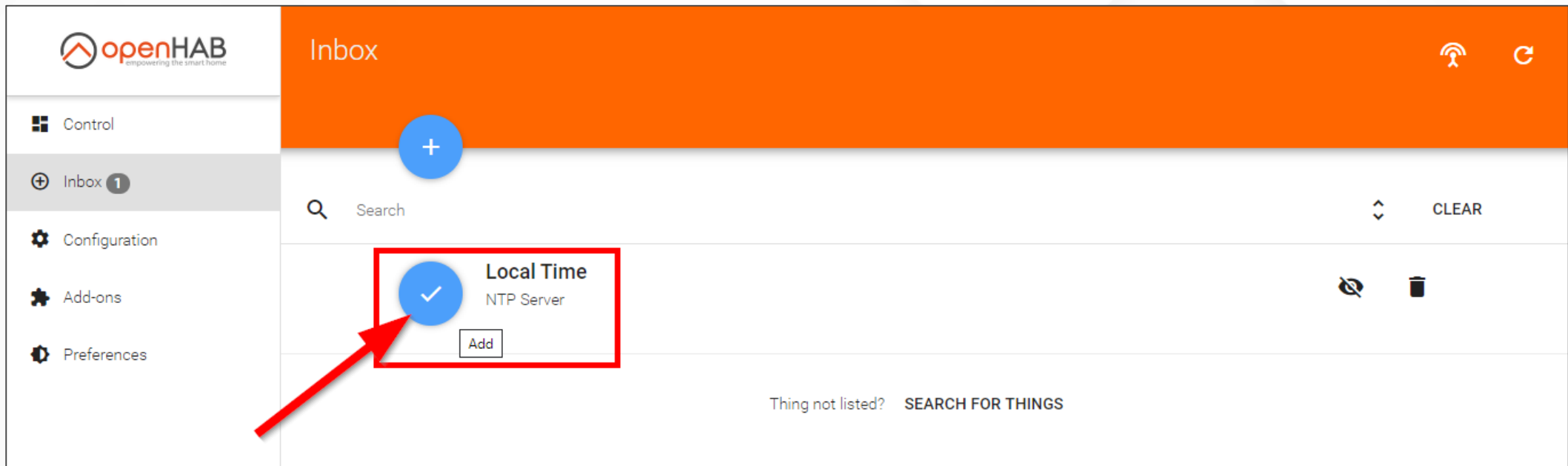
Sitemaps

Importar el binding NTP.



Sitemaps

Crear la thing correspondiente a través de Inbox.



Sitemaps

Cada elemento (item) debe crearse con la siguiente sintaxis

tipoElemento
nombreElemento
"Descripción" <icono> {
canal }

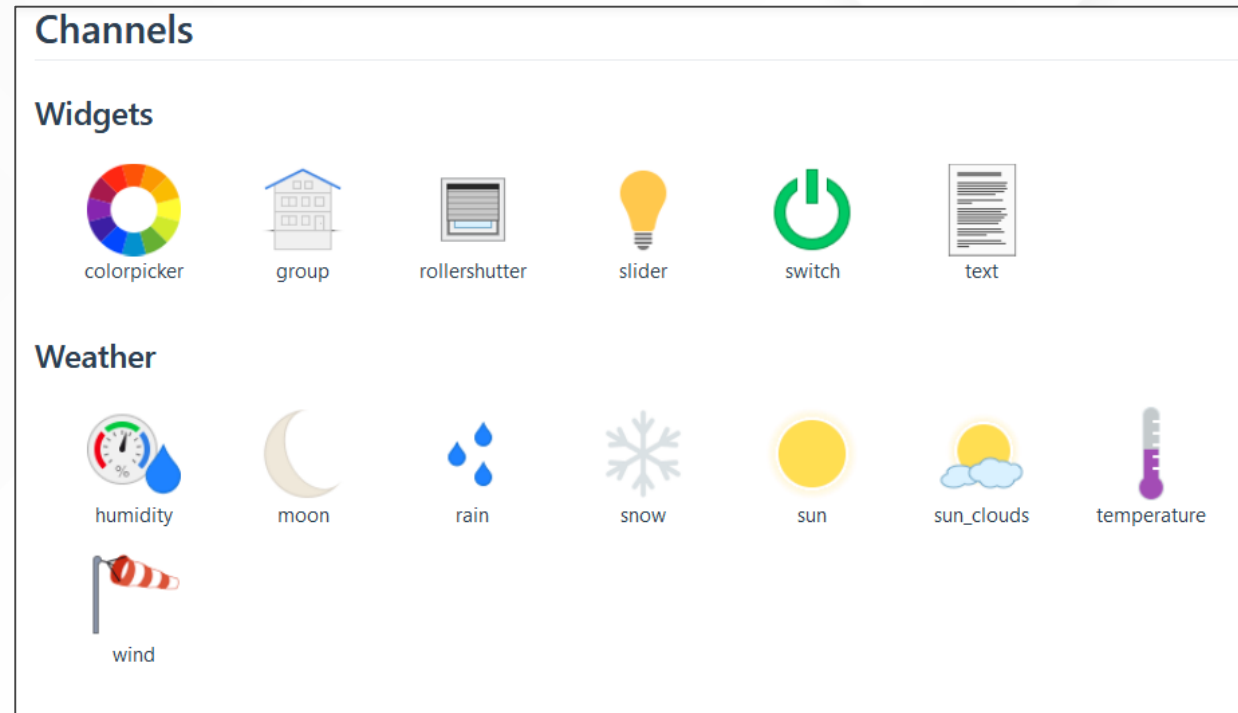
Donde tipoElemento puede ser cualquiera de la tabla.

Item Name	Description	Command Types
Color	Color information (RGB)	OnOff, IncreaseDecrease, Percent, HSB
Contact	Item storing status of e.g. door/window contacts	OpenClosed
DateTime	Stores date and time	-
Dimmer	Item carrying a percentage value for dimmers	OnOff, IncreaseDecrease, Percent
Group	Item to nest other Items / collect them in Groups	-
Image	Holds the binary data of an image	-
Location	Stores GPS coordinates	Point
Number	Stores values in number format, takes an optional dimension suffix	Decimal
Number:<dimension>	like Number, additional dimension information for unit support	Quantity
Player	Allows to control players (e.g. audio players)	PlayPause, NextPrevious, RewindFastforward
Rollershutter	Typically used for blinds	UpDown, StopMove, Percent
String	Stores texts	String
Switch	Typically used for lights (on/off)	OnOff

Sitemaps

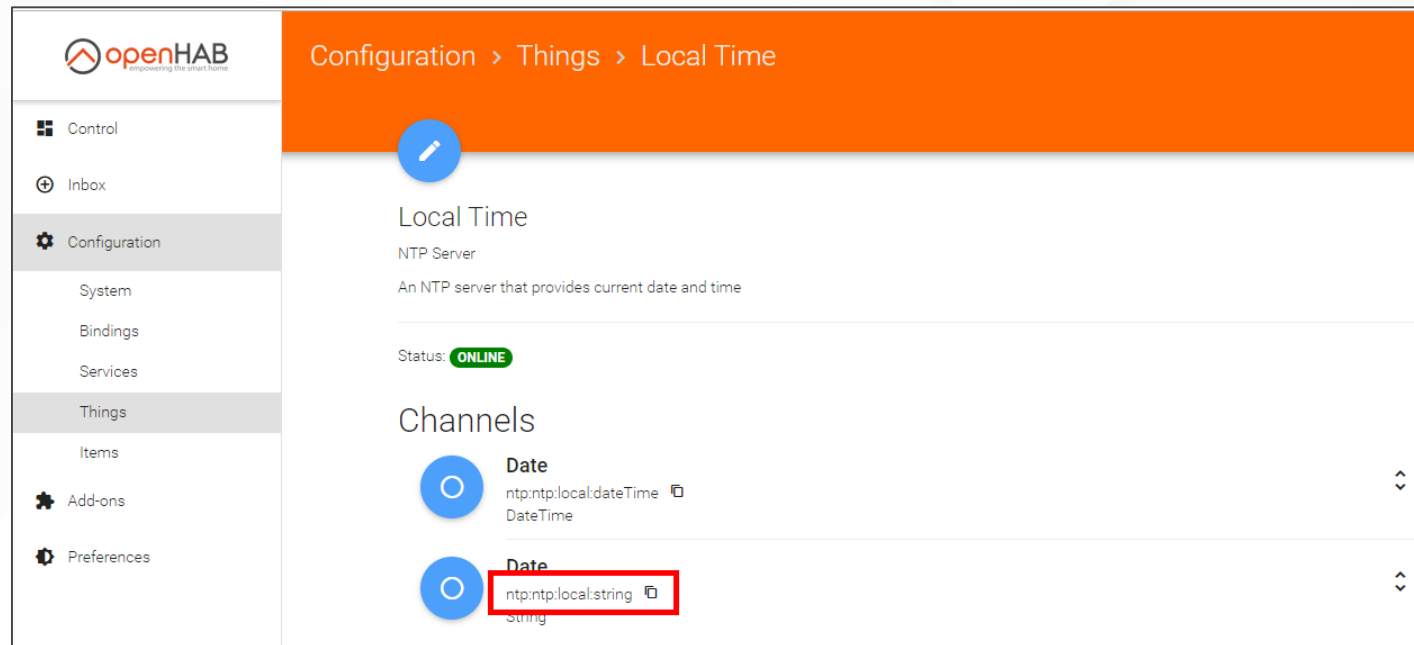
<icono> puede ser cualquiera de esta lista

<https://www.openHAB.org/docs/configuration/iconsets/classic/>



Sitemaps

Y **canal** es el nombre del canal que vincula la thing con el item, y que podemos obtener en la configuración de la thing



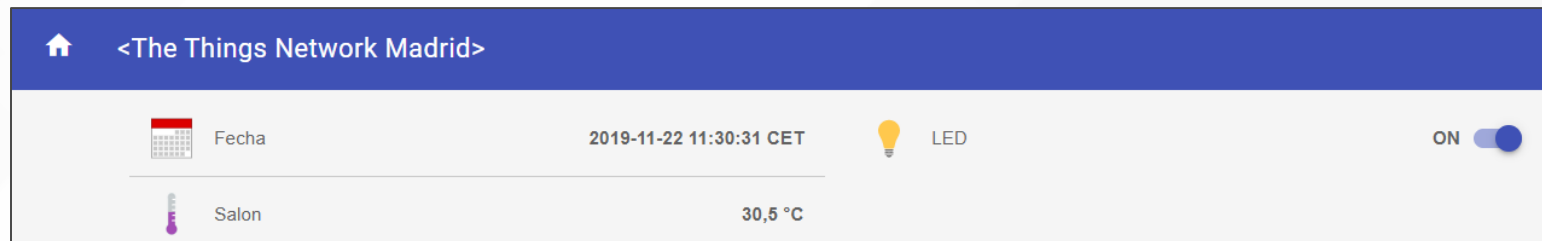
Sitemaps

Crear el archivo C:\openHAB\conf\items\miselementos.items con el siguiente contenido:

```
String Fecha_NTP "Fecha" <calendar> { channel="ntp:ntp:local:string" }
```

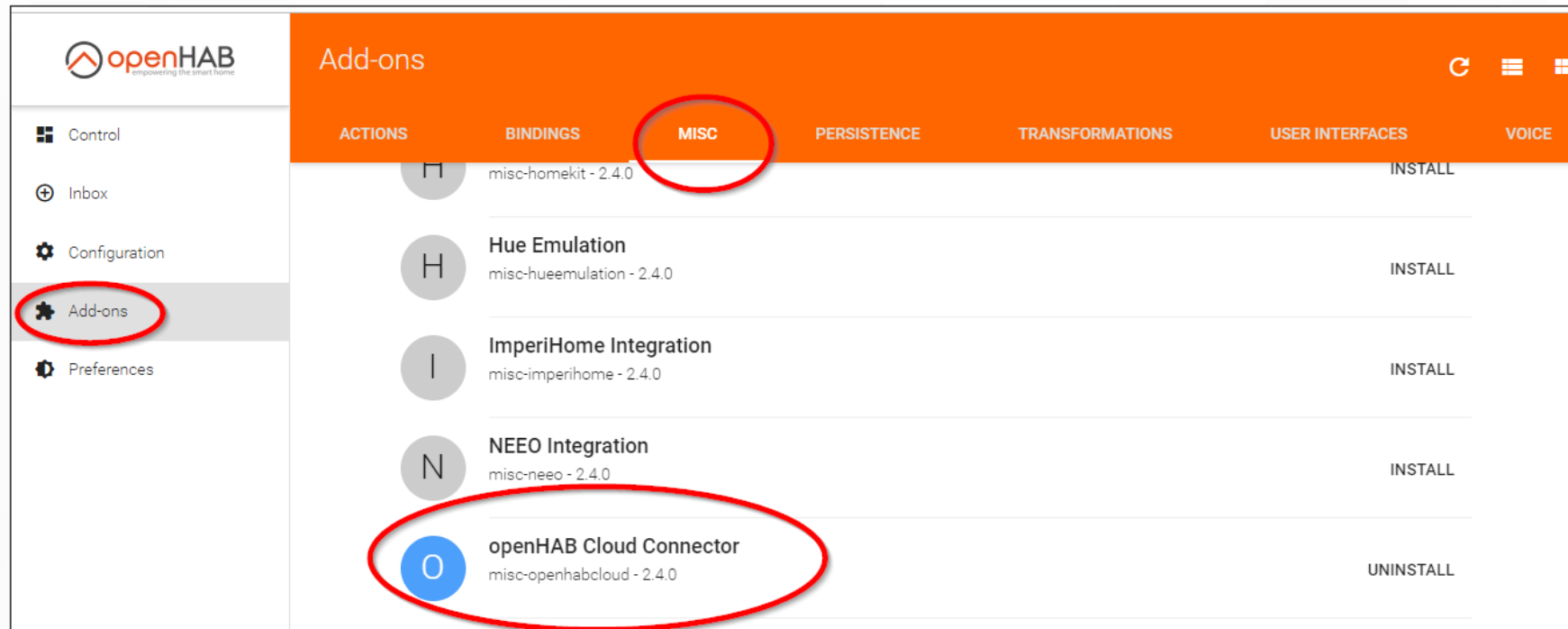
Y modificar el archivo ttn.sitemap:

```
sitemap ttn label="<The Things Network Madrid>" {  
  Text item=Fecha_NTP label="Fecha"  
  Switch item=LED label="LED"  
  Text item=Temperatura label="Salon [%1f °C]" icon="temperature"  
}
```



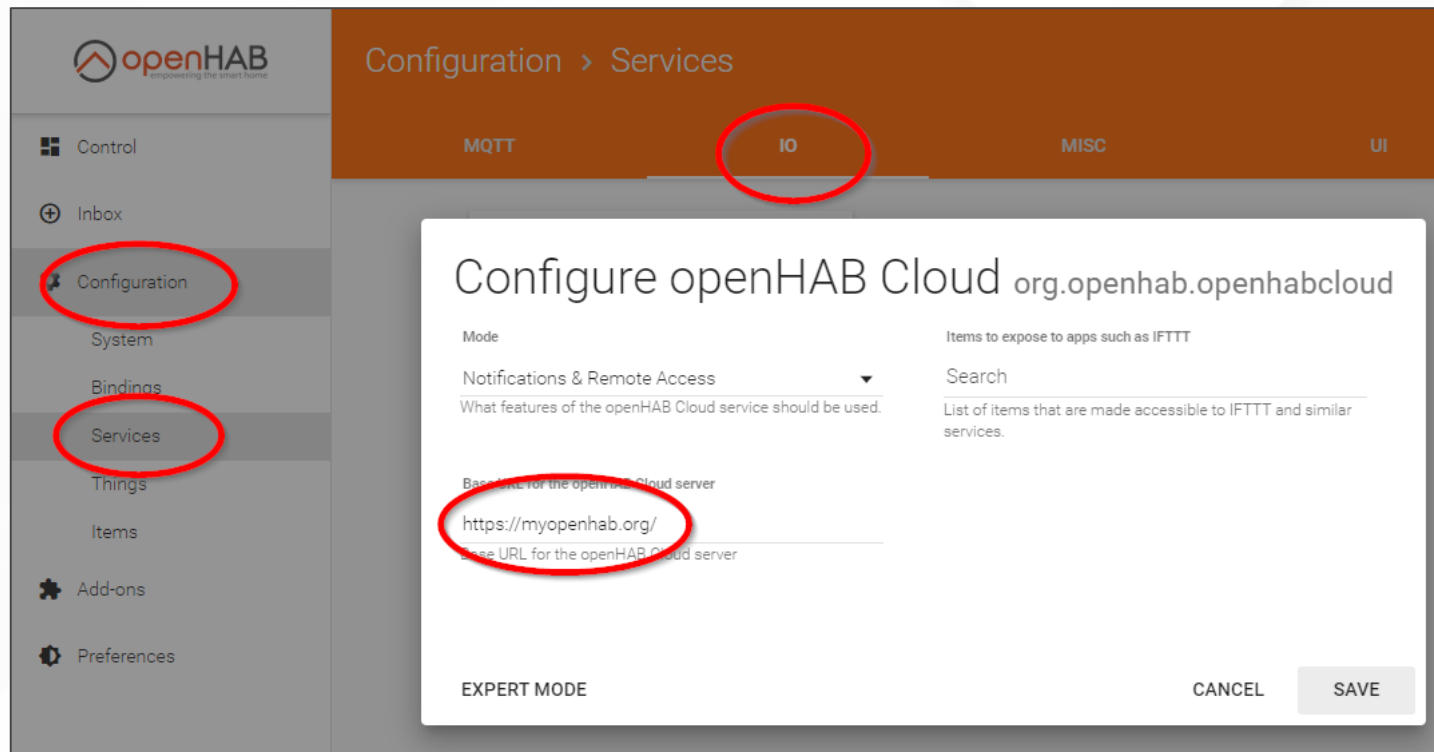
App móvil

Instalar el Add-on openHAB Cloud Connector



App móvil

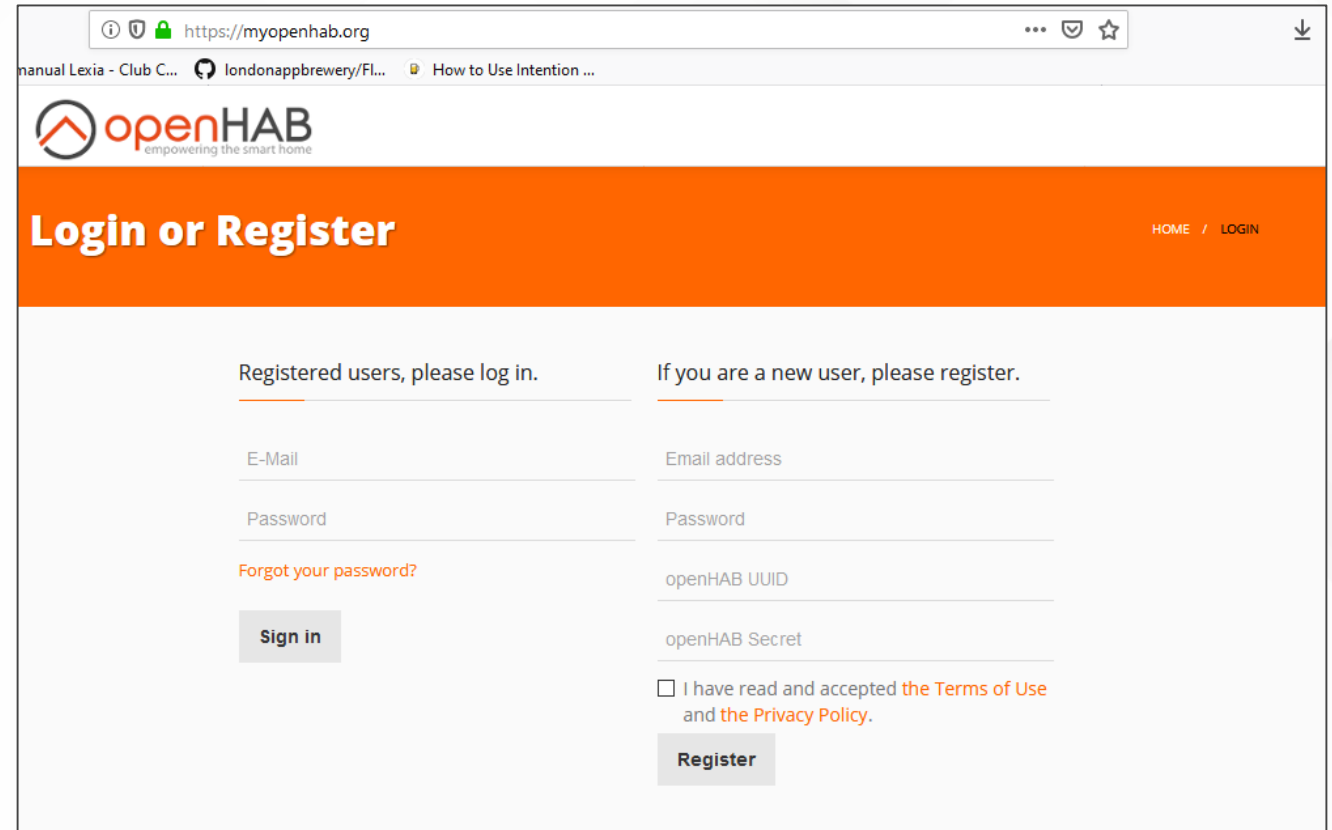
Configurar openHAB Cloud Connector para que utilice el servidor <https://myopenhab.org>



App móvil

Crear una cuenta en
<https://myopenhab.org>

- UUID:
C:\openHAB\userdata\uuid
- Secret:
C:\openHAB\userdata\openhabcloud\secret



The screenshot shows the web interface of myopenhab.org. The browser address bar displays 'https://myopenhab.org'. The page features the openHAB logo with the tagline 'empowering the smart home'. Below the logo is an orange banner with the text 'Login or Register' and links for 'HOME' and 'LOGIN'. The main content area is divided into two columns. The left column, titled 'Registered users, please log in.', contains input fields for 'E-Mail' and 'Password', a link for 'Forgot your password?', and a 'Sign in' button. The right column, titled 'If you are a new user, please register.', contains input fields for 'Email address', 'Password', 'openHAB UUID', and 'openHAB Secret'. It also includes a checkbox for 'I have read and accepted the Terms of Use and the Privacy Policy.' and a 'Register' button.

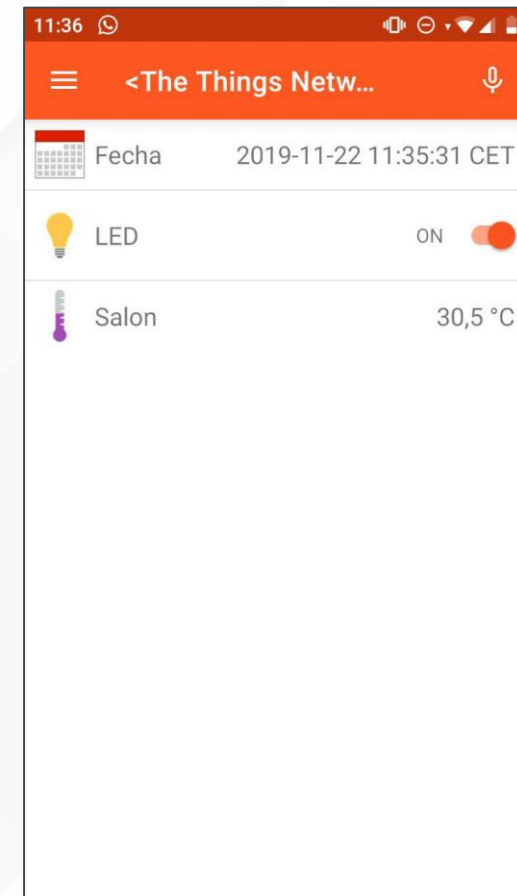
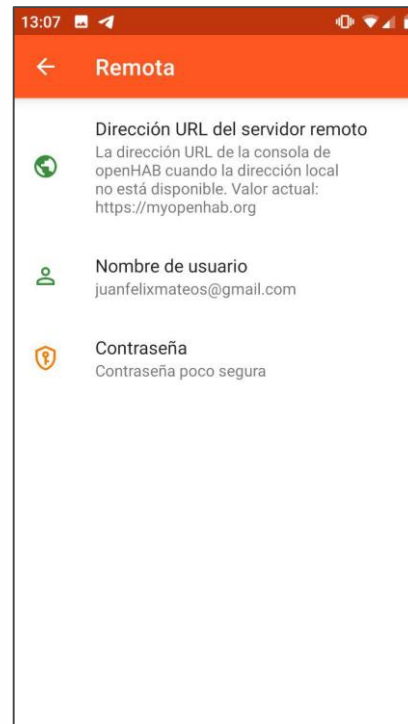
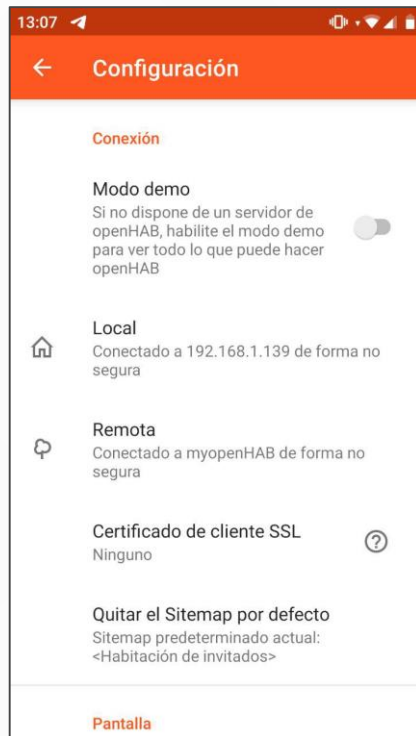
App móvil

Instalar la aplicación en el móvil



App móvil

Configurar la app para que se conecte a myopenhab.org

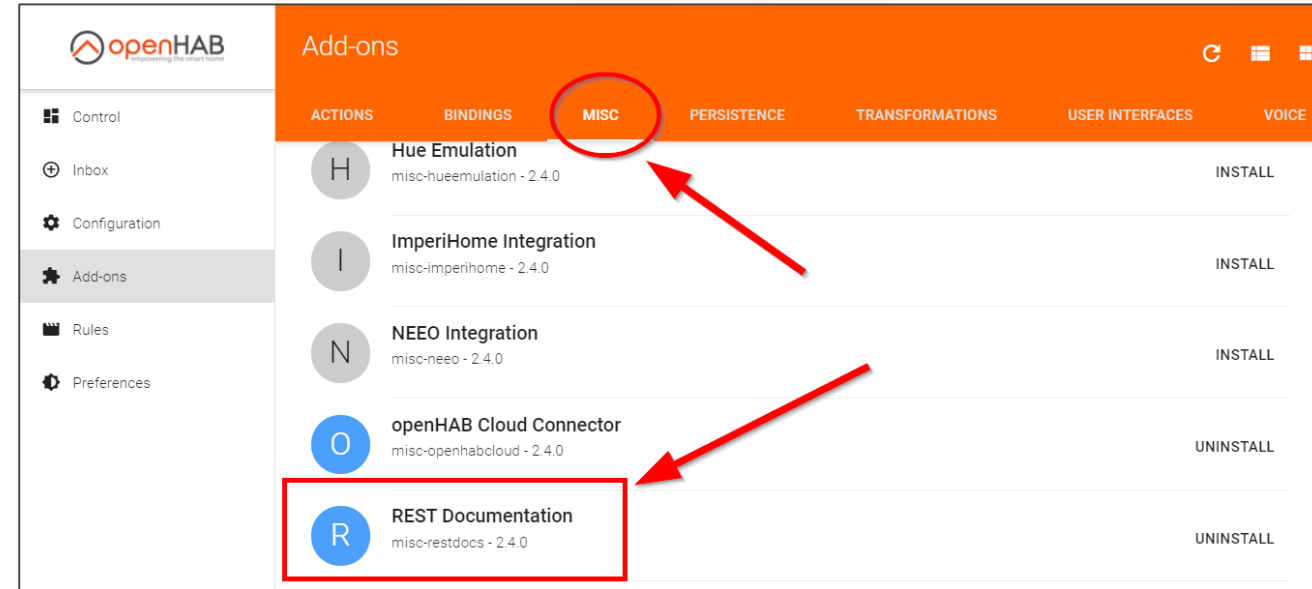


Alexa

Para que Alexa pueda identificar a nuestro nodo necesitamos aportar a los items **metainformación** que indique qué posibilidades ofrecen.

Para aportar esta metainformación vamos a recurrir a la Rest API de openHab que, si instalamos el binding **openHAB Rest Documentation**, está disponible a través de Swagger en:

- <http://localhost:8080/doc/index.html>



Alexa

El listado de metadatos está disponible en:

- <https://www.openhab.org/docs/ecosystem/alexa/#supported-item-metadata>
- Por ejemplo:
 - Interruptor: `PowerController.powerState`
 - Sensor: `TemperatureSensor.temperature`

Alexa

Metadatos de LED

PUT `/items/{itemname}/metadata/{namespace}` Adds metadata to an item.

Parameters

Parameter	Value	Description	Parameter Type	Data Type
itemname	LED	item name	path	string
namespace	alexa	namespace	path	string
body	<code>{"value": "PowerController.powerState"}</code>	metadata	body	Model Model Schema

Parameter content type: application/json

```
{  
  "value": "string",  
  "config": {}  
}
```

Click to set as parameter value

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		
201	Created		
400	Metadata value empty.		
404	Item not found.		
405	Metadata not editable.		

Try it out!

[Hide Response](#)

Alexa

Podemos comprobar que el metadato se ha añadido correctamente revisando el archivo:

- C:\openHAB\userdata\jsondb\org.eclipse.smarthome.core.items.Metadata.json

```
28  "alexa:LED": {  
29      "class": "org.eclipse.smarthome.core.items.Metadata",  
30      "value": {  
31          "key": {  
32              "segments": [  
33                  "alexa",  
34                  "LED"  
35              ]  
36          },  
37          "value": "PowerController.powerState",  
38          "configuration": {}  
39      }  
40  }
```

Alexa

Metadatos de Temperatura

PUT

/items/{itemname}/metadata/{namespace}

Adds metadata to an item.

Parameters

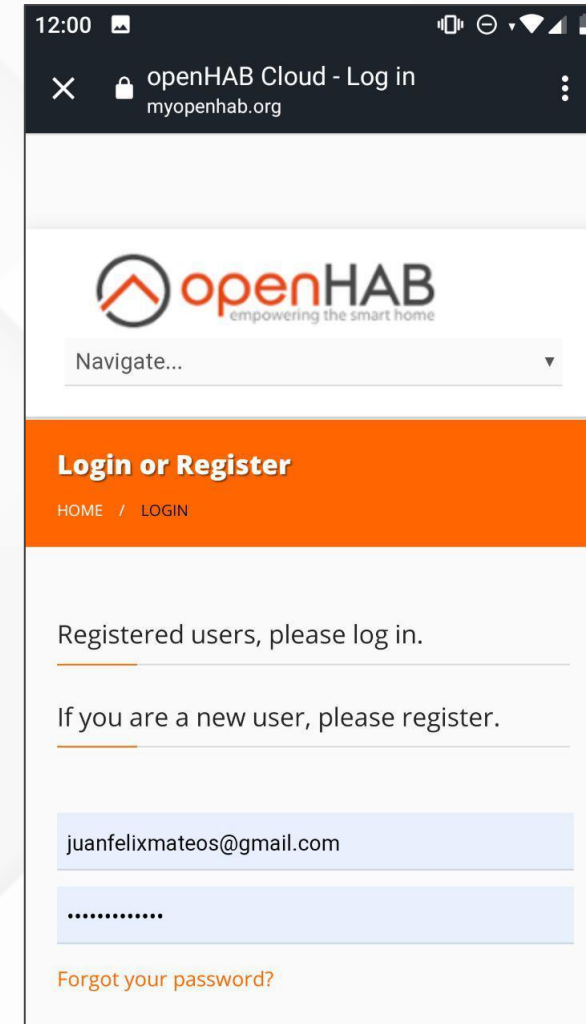
Parameter	Value	Description	Parameter Type	Data Type
itemname	<input type="text" value="Temperatura"/>	item name	path	string
namespace	<input type="text" value="alexa"/>	namespace	path	string
body	<div><pre>{"value": "TemperatureSensor.temperature", "config": {"scale": "Celsius"}}}</pre><div>Parameter content type: <input type="text" value="application/json"/></div></div> <td>metadata</td> <td>body</td> <td><div>ModelModel Schema</div><div><pre>{ "value": "string", "config": {} }</pre><div>Click to set as parameter value</div></div></td>	metadata	body	<div>ModelModel Schema</div> <div><pre>{ "value": "string", "config": {} }</pre><div>Click to set as parameter value</div></div>

Response Messages

HTTP Status Code	Reason	Response Model	Headers
200	OK		

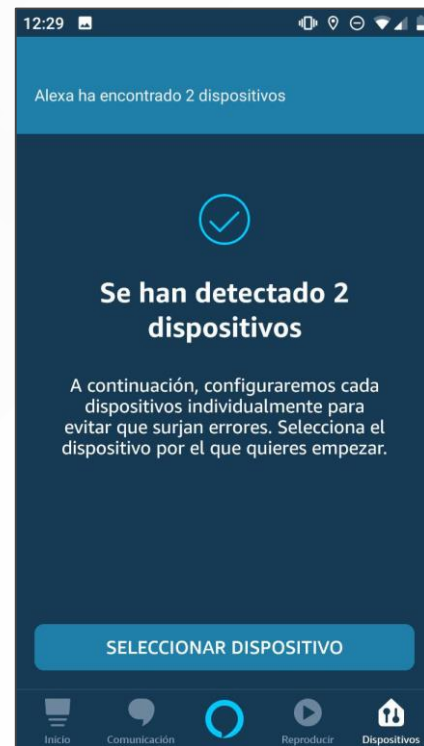
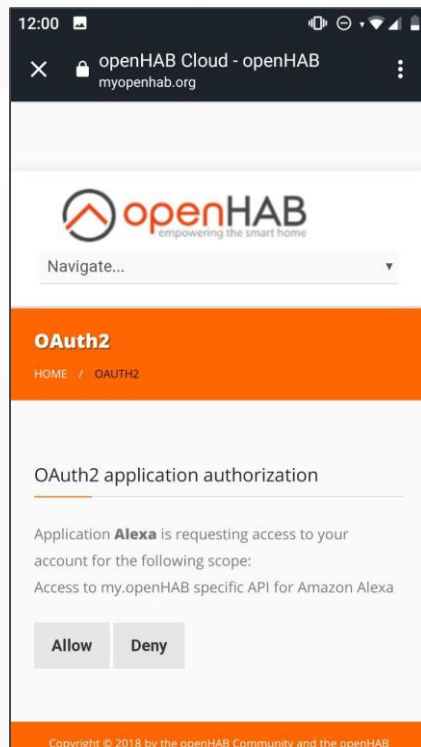
Alexa

- Instalar la app Alexa en el móvil
- Activar el skill oficial de openHAB
- Acreditarse en myopenhab



Alexa

- Aceptar la autorización OAuth2
- Permitir que Alexa detecte nuestros 2 dispositivos



Alexa

Interactuar con los siguientes comandos de voz:

- ¿Cuál es la temperatura de Temperatura?
- Activa LED
- Desactiva LED
- ¿Cuál es el estado de LED?
 - Todavía no está disponible ¿o sí?
 - <https://github.com/openhab/openhab-alexa/issues/239>

jfmateos@educa.madrid.org
juanfelixmateos@gmail.com

GRACIAS

