

# JA-SmartHub Manual



## Chapters

1. Conditions
2. Connections
3. Commissioning / connecting step-by-step plan
4. Approach web-page
5. A. Walk through the 'Flow'  
B. Example statuses  
C. IFTTT Connection
6. IFTTT Applets configuration
7. Continue configurations
8. Debugging
9. Change settings of existing Applets.

## 1. Conditions

There are a number of conditions for using the JA-Smarthub in combination with IFTTT.

### **Needed for correct installation:**

- Passwords of all products to be linked (like Sonos, Hue, etc);
- LAN connection active in Smarthub + working internet connection;
- Linked to Jablotron control panel via BUS;
- IFTTT account created for end user.

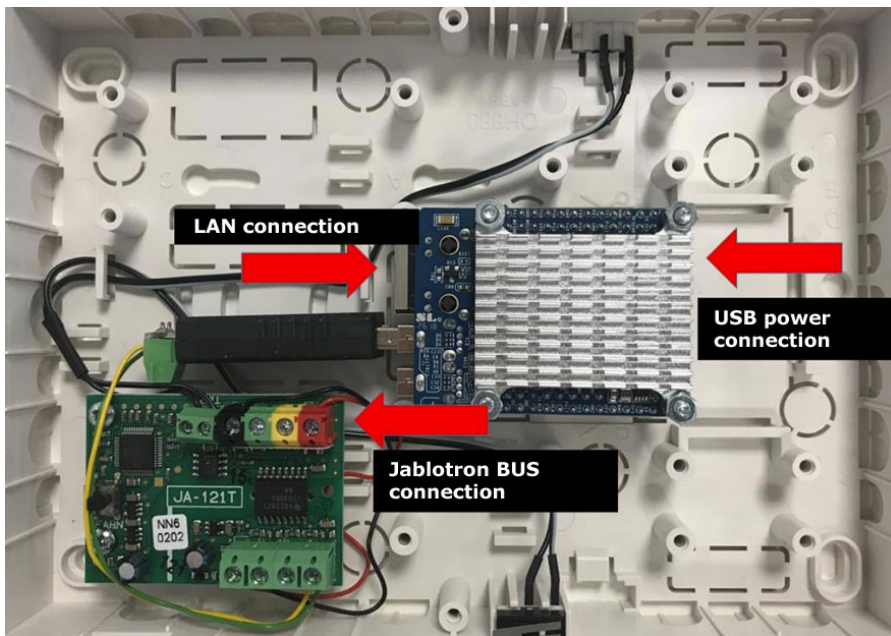
It is not possible to use one IFTTT account for multiple SmartHubs.

In addition, it is not possible to create an engineer account with IFTTT. So create an account for the end user for each installation.

## 2. Connections

The JA-Smarthub has 3 connections:

- The lan connection for the network.
- The usb power supply for the HUB.
- The Jablotron bus connection for the RS485 module.



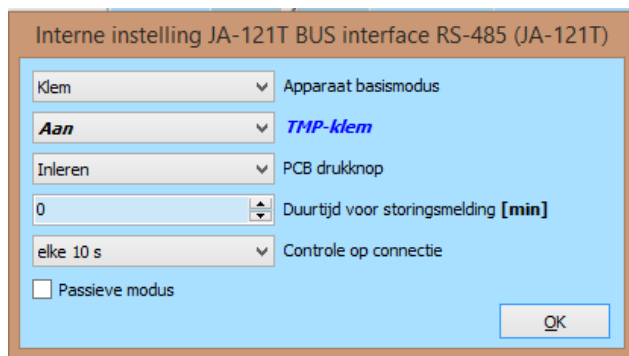
### 3. Commissioning / connecting Step-by-Step plan

1. Connect the bus to the RS485 module
2. Connect the Lan
3. Connect the usb power

Enrolling the RS485 module in the Jablotron control panel

If the module is connected, the module can be enrolled with F-link.

The following settings must be made in the module (the TMP terminal ensures that the tamper of the lid and mounting tray is signaled). Connection check ensures that an OK message is received in the JA-Smarthub every 10 seconds.



### 4. Approach web-page

After connecting the Ja-Smarthub you can open an internet browser and navigate to the next page <http://ja-hub.local:1880>

Login to the page with the following information:

**UserName** = admin  
**Password** = Jablotron (note the capital letter J)

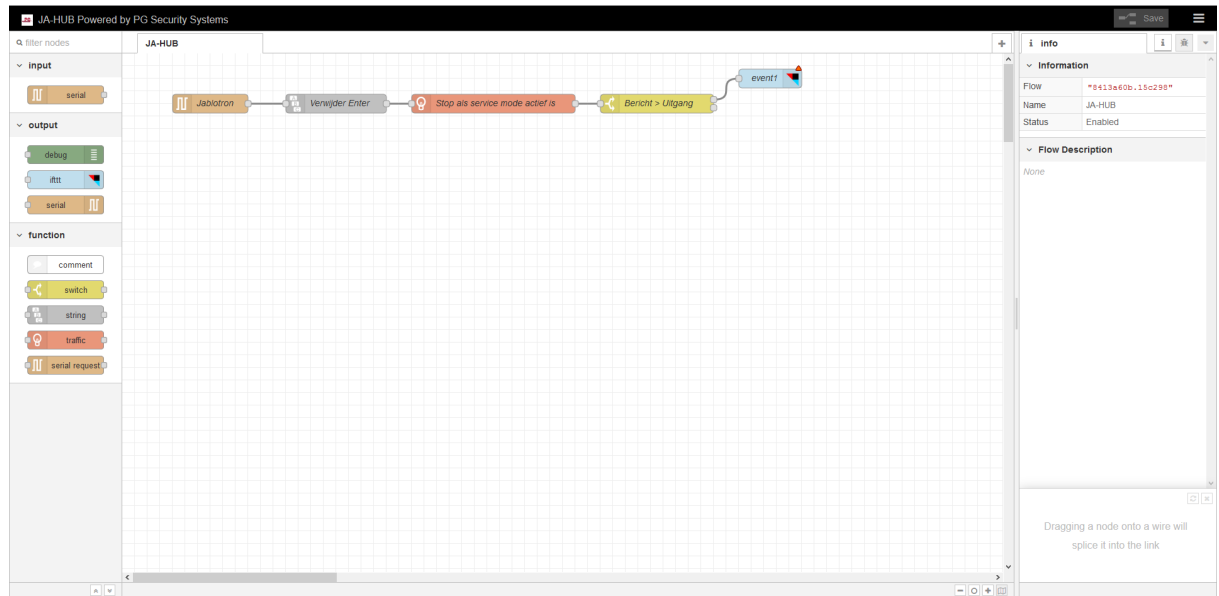
Username:  
|

Password:

Login

After logging in, a cookie is created for 7 days, so you no longer have to log into the page for the next 7 days.

The following screen appears after logging in. This is enlarged on the following page.



filter nodes

input



output



function



JA-HUB

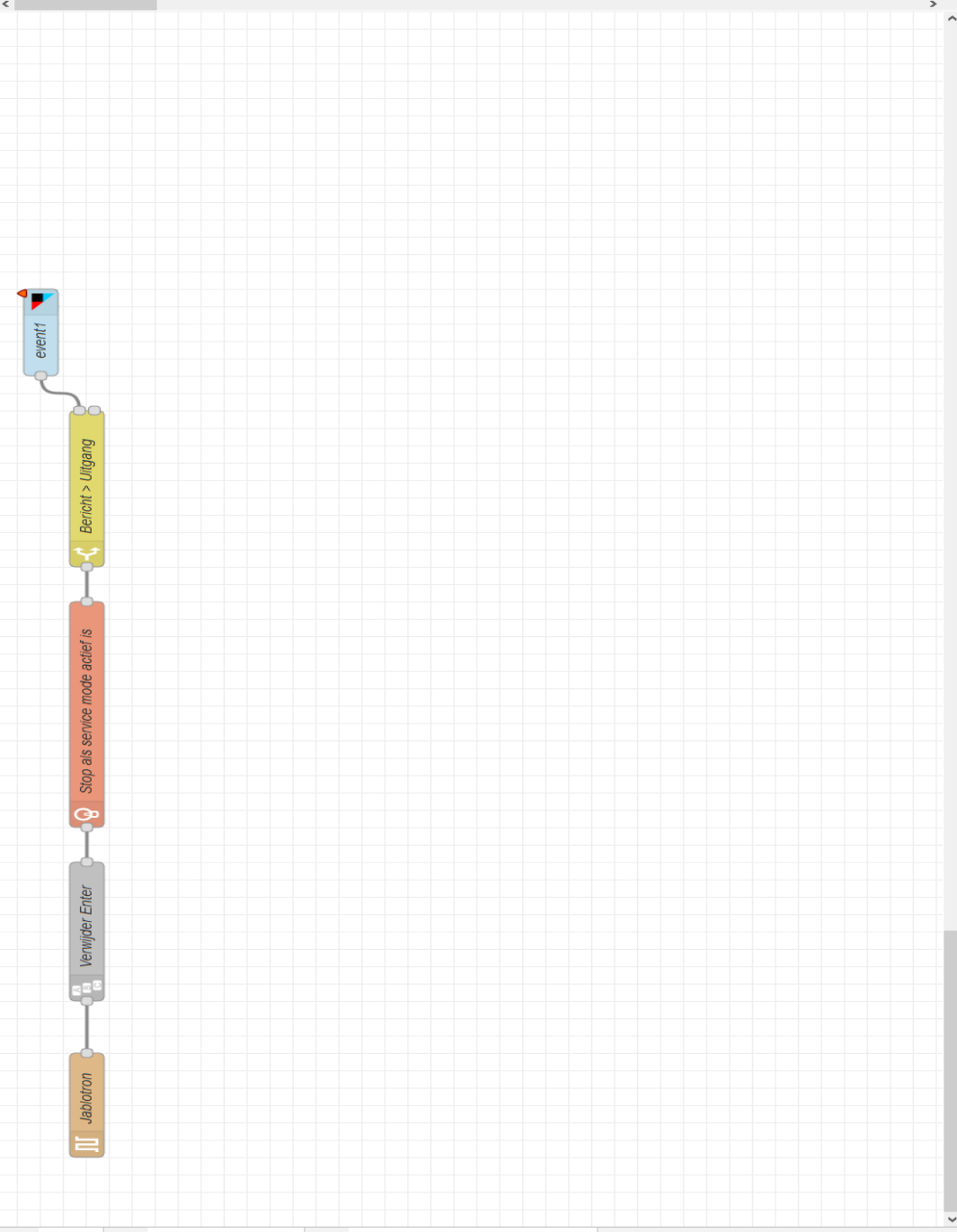
i info

Information

Flow	*e413a60b.15c298*
Name	JA-HUB
Status	Enabled

Flow Description

None

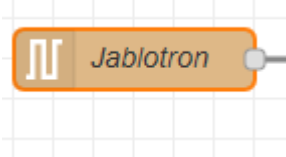


Dragging a node onto a wire will splice it into the link

## 5. A. Walk through the "Flow"

We go through the "FLOW" from left to right.

The first block is the RS-485 block which connects to the Jablotron control panel.



This block "listens" to the Jablotron panel and captures the messages that come from the control panel.

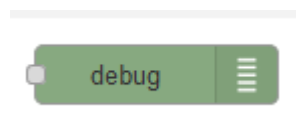
For example, if a section turns on or off, a message arrives at the Jablotron block. An example of this is for example:

STATE X READY = Section X switched off

STATE X ARMED= Section X switched on

If we want to see all messages sent from the Jablotron control panel, we can add a debug block to the flow.

On the left we see the tree with all the blocks that are available. We choose the debug block there and drag it onto the window.

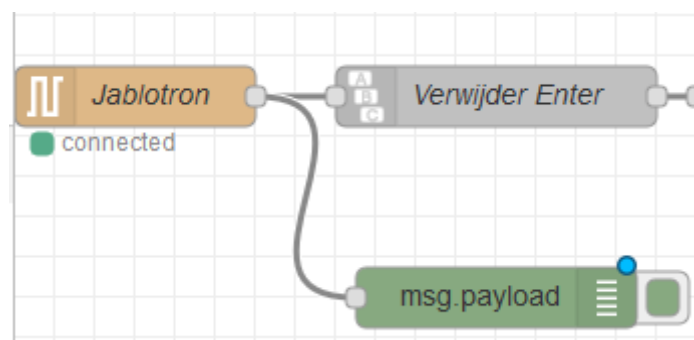


Next, we must "link" this block to the flow.

If we click on the Jablotron block on the gray block with the left mouse button and hold it, we can draw a line to the gray block on the debug block.

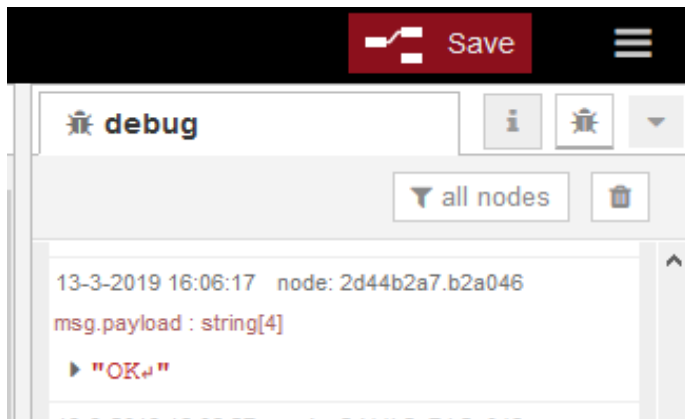
By clicking on the green block on the right side of the block, the debug can be switched on and off.

When we are done debugging it is wise to remove the block.

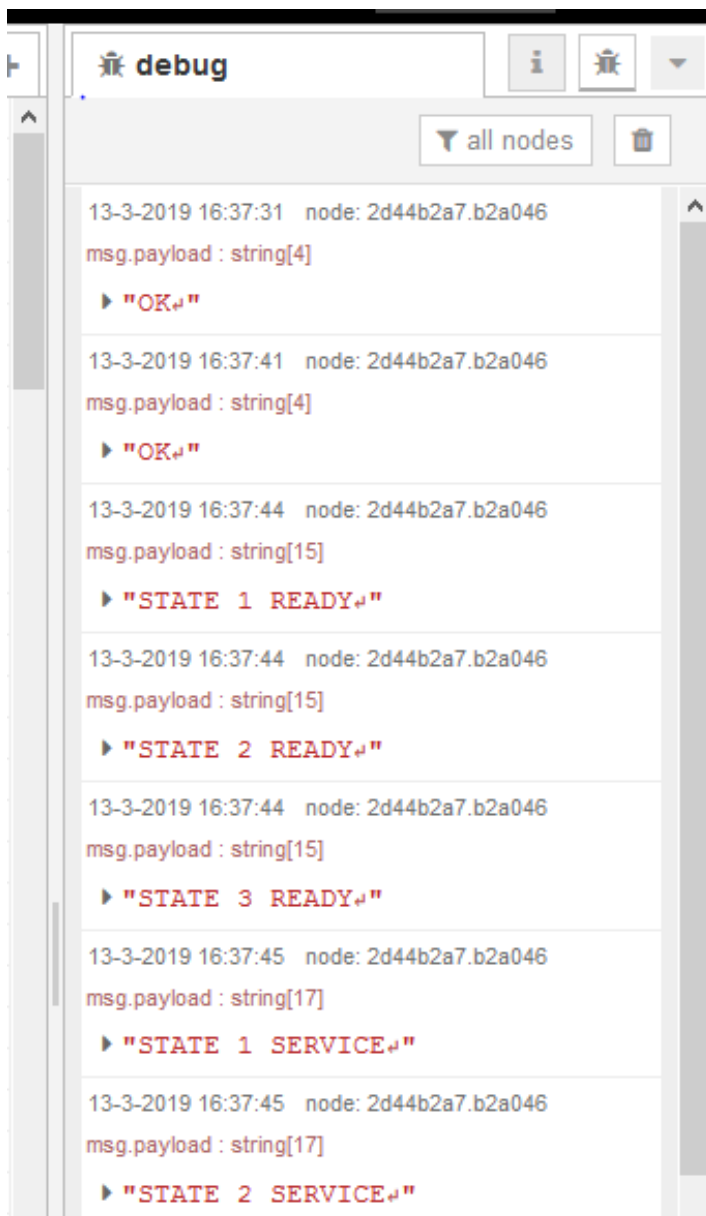


Now that both are connected, we have to click on save in the top right to implement the changes.

Then click on the icon  to see the debug window:



Then we can see on the right side all messages that come through the Jablotron block.



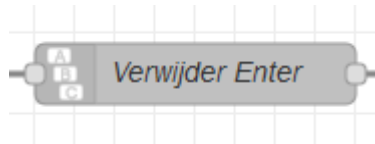


For example, we can extract the following statuses from this:

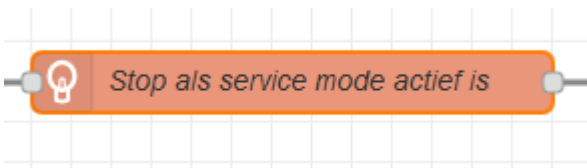
## 5.B Example statuses

PG X ON = PG X output is on  
PG X OFF =PG X output is off  
STATE X READY = Section X unset  
STATE X ARMED= Section X set  
ENTRY X ON = Exit delay of Section X is active  
ENTRY X OFF=Exit delay of Section X is not active  
EXIT X ON = Exit delay of Section X is active  
EXIT X OFF=Exit delay of Section X is not active

The next block removes the enter character that comes after each message, nothing needs to be configured.



After this comes the block:



This block ensures that when the exchange is in service mode, no more messages are forwarded. And the controls are therefore not carried out. Also nothing has to be configured on this block.

The next block is the "MESSAGE > OUTPUT" block.

In this block we look at the content of the message and switch a software output on it:



If we click on this block, the following window will appear.

## Edit switch node

Delete

Cancel

Done

### node properties

Name

Bericht > Uitgang

Property

msg.payload

≡	==	▼	▼ a_z	STATE 1 ARMED	→ 1	✕
≡	==	▼	▼ a_z	STATE 1 READY	→ 2	✕
≡	==	▼	▼ a_z	PRFSTATE 08000000000000000000	→ 3	✕
≡	==	▼	▼ a_z	PRFSTATE 00000000000000000000	→ 4	✕
≡	==	▼	▼ a_z	PG 7 ON	→ 5	✕
≡	==	▼	▼ a_z	PG 7 OFF	→ 6	✕

+ add

checking all rules

recreate message sequences

### > node settings

If we look at the contents of the block, we see that as the message "STATE 1 ARMED" from the Jablotron central unit the first output is switched.

So if we want an action to happen when section 1 turns on, we have to connect the IFTTT block to the 1st output.

For example, if we want an action to happen when we turn off the 1st section, then we must connect to the 2nd exit port.

### 5.C. IFTTT Connection

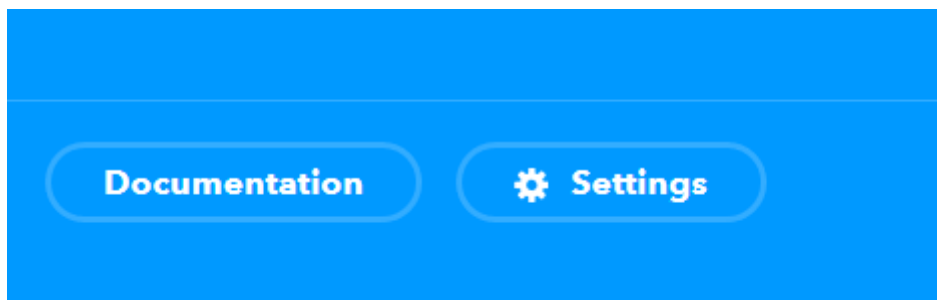
The next block in the flow is the IFTTT block.

Here we have to enter our unique API key once, to retrieve this key we need an account for IFTTT, so a unique account must be created for each customer.

To do this, go to IFTTT.com and create an account.

Then log in to IFTTT.com with the account you just created, open a new tab in the browser and go to [maker.ifttt.com](https://maker.ifttt.com).

Then click on "Settings":



### Account Info

Connected as:

URL: <https://maker.ifttt.com>

/use/jREWvc5

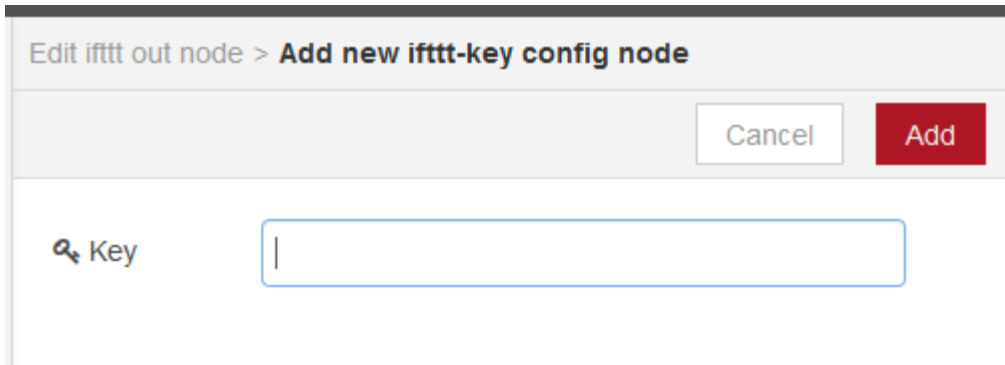
55WR4

Status: **active**

**Edit connection**

Now copy the value after use/

Then open the IFTTT block in the JA-Smarthub.

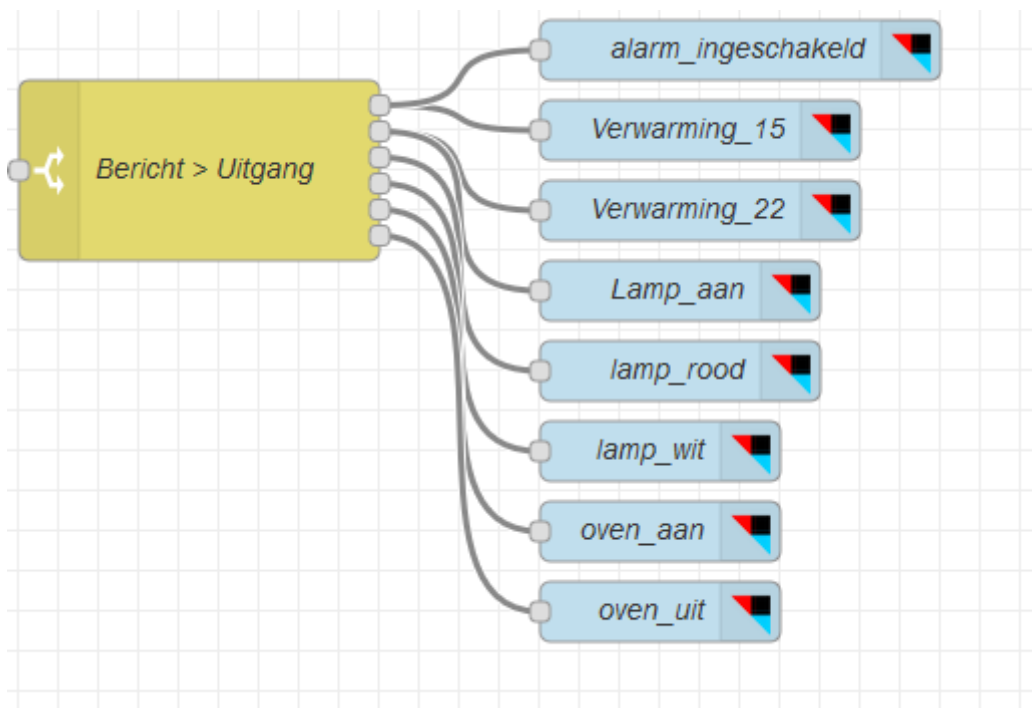


Paste the unique API key here.

Press add to save the key. Then we can enter a self-chosen value in Event Name. For example alarm\_enabled (we can choose here which texts we use).

We will use this Event name later in IFTTT to set a webhook trigger.

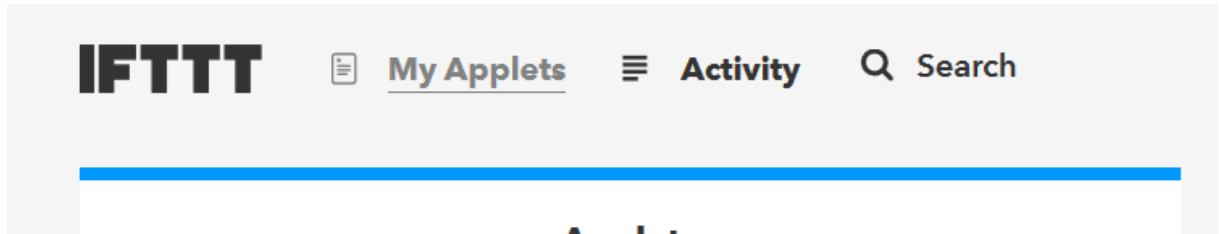
Connect the first output block of MESSAGE> OUTPUT as shown below:



So we can have multiple actions performed simultaneously by connecting multiple IFTTT blocks with the same output.

## 6 IFTTT Applets

Next, we want an action to happen in IFTTT if we send the alarm\_enabled message. To do this, go to IFTTT.com and click on My Applets, and then on new applet.

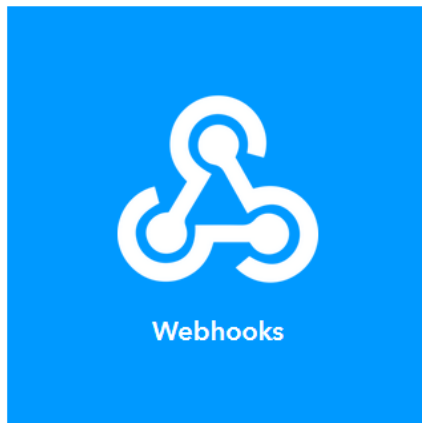


**New Applet**

Now click on the blue + this, at "choose a service" we choose webhook

## Choose a service

Step 1 of 6



In the next window, enter the event text that we entered in the JA-Smarthub in the IFTTT block



## Complete trigger fields

Step 2 of 6

### Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

#### Event Name

The name of the event, like "button\_pressed" or "front\_door\_opened"

**Create trigger**

Then click: create trigger.

**if**  **then**  **that**

Now we can click on that and we can choose an action that we want to perform.

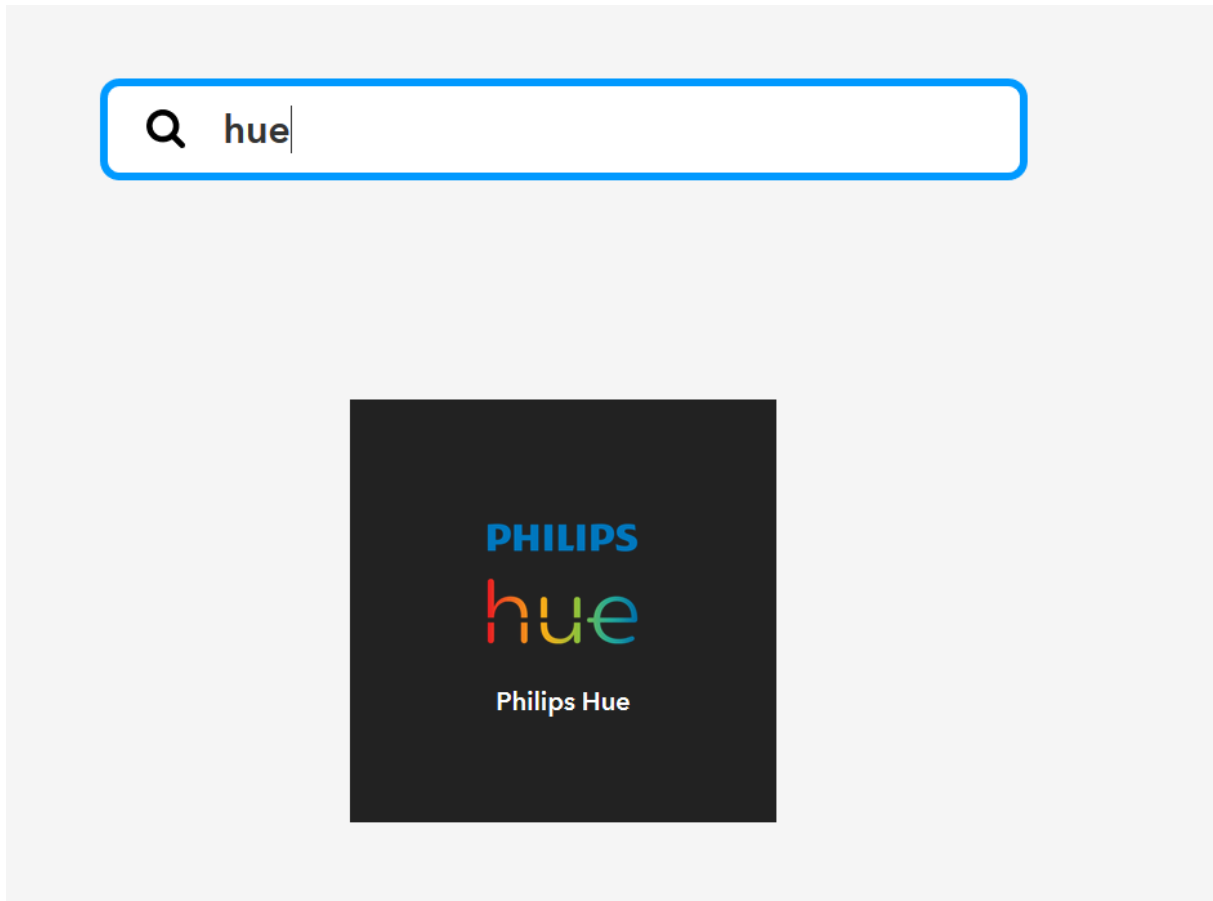
## 7. Continue configurations

Every operation that must be performed by an IFTTT-linked product has its own Applet. In accordance with the steps taken above, these can be built one by one.

To know if a product can be linked via IFTTT, navigate to:

<https://ifttt.com/services>

Then we can search on the product in the example below we search on Philips Hue:



When we click on the service, we see all sorts of various applets that have already been created by people. We now go all the way to the bottom of the page where we find triggers and actions:

### Philips Hue triggers and actions

---

We can click on that and we will see all the possible actions that we can do with actions.



## Triggers

There are currently no triggers for this service.



## Actions

### Set a scene in a room

This Action will set a scene on your hue lights.  
NOTE: in order to add lights to a room, you must be using the newest version of the Hue app

#### Action Fields

Which scene?

### Turn on lights

This Action will turn on your hue lights.

#### Action Fields

Which lights?

### Turn off lights

This Action will turn off your hue lights.

So we see that we can perform various actions, depending on the chosen service, various actions are possible.

## 8. Debugging

If we look at the IFTTT webpage or app we also see a “Activity” heading.

Here we can see all the actions that have been carried out and whether they have been successful and what went wrong.





### Applet ran

Mar 18 - 2:06 PM



If Maker Event "Verwarming\_15", then Set temperature for Office Thermostat

The event named "Verwarming\_15" occurred on the Maker Webhooks service

▼ Show details



### Applet ran

Mar 18 - 2:06 PM



If Maker Event "Lamp\_Uit", then Turn off All

The event named "lamp\_uit" occurred on the Maker Webhooks service

▼ Show details

## 9. Change settings of existing Applets.

If we want to change an existing IFTTT link, we click on "My Applets".



Then we see the overview with all Applets that we have created, now click on the applet that needs to be changed. We can then click on the gear symbol to see and change the settings.

**My Applets > Webhooks**

