REAL SMART HOME

REAL SMART HOME GmbH

APPMODULE

IFTTT Connect

Smart Home App Documentation

Version: 1.1.0
Type: Application
Article No.: BAB-089

Documentation version II
Actual state 09/2021
Date: 21. September 2021

EN

REAL SMART HOME GmbH

Hörder Burgstraße 18
D-44263 Dortmund

Email: info[at]realsmarthome.de

Fon: +49 (0) 231-586 974 -00
Fax: +49 (0) 231-586 974 -15
www.realsmarthome.de

TABLE OF CONTENTS

1 Introduction 4
Important information on the operating instructions 4
2 IFTTT Connect Functional overview 5
3 The innovative, modular Smart Home App concept for the building automation 6
3.1 Information about the APPMODULE 6
4 Smart Home App installation / update 7
5 Smart Home App Settings 8
5.1 IFTTT Connect 8
5.2 Telegram Trigger for IFTTT Maker Event 8
5.3 Values to include (optional) 10
5.4 External address of the APPMODULE (for Webhooks) 11
6 Application example 14
6.1 Instruction: Location (GEOFENCE) trigger KNX group address 14
Configuration in **APP**MODULE 14
Configuration in IFTTT Webinterface 17
Testing the function 24
6.2 Instruction: Telegram value from KNX to Google Drive storage 26
Configuration at IFTTT Webinterface 26
Configuration **APP**MODULE 31
Testing the function 36
7 Attachment 39
7.1 Datapoint Types 39

1 INTRODUCTION

Thank you for your trust, and the purchase of the **IFTTT Connect** -Smart Home App for the BAB **APP**MODULE. With **IFTTT Connect** - Smart Home App you connect all functions of the **APP**MODULE with the web applications
of ifttt.com. Transfer KNX counter readings to Google tables and much more.

This documentation will help you get started with the Smart Home App and aims to improve your setup experience.

REAL SMART HOME GmbH

IMPORTANT INFORMATION ON THE OPERATING INSTRUCTIONS

We reserve the right continually improve the product. This entails the possibility that parts of this documentation might be out-of-date. You will find the latest information at:

www.bab-appmarket.de

This Smart Home App is an independent product, with no legal ties to IFTTT Inc. Neither **BAB** APPMARKET GmbH nor the developer of this Smart Home App take any claim in the trademarks owned by IFTTT Inc.

2 IFTTT CONNECT FUNCTIONAL OVERVIEW

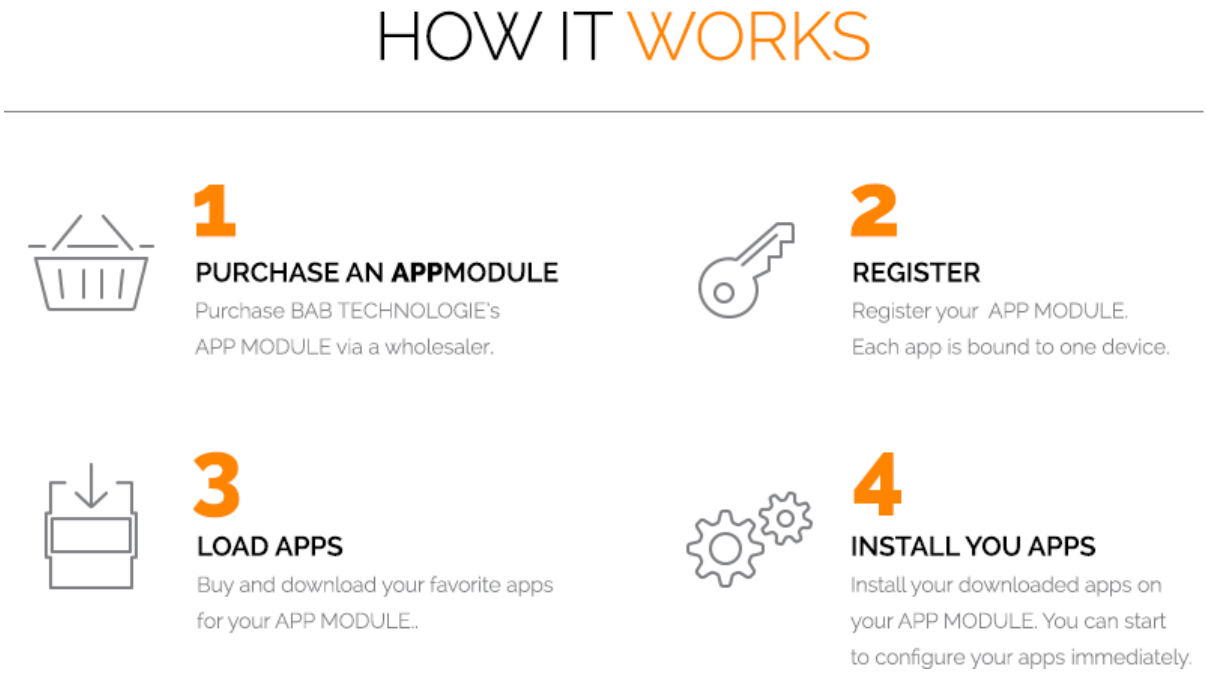
With this app for the **APP**MODULE you connect over 600 IoT and web services with KNX or EnOcean. For example, transfer KNX counter readings to Google Tables or link your current geographical location to KNX scenes.

Highlights

- Connects over 600 IoT services with KNX, up to 10 bidirectional connections
- Convenient URL generator
- Flexible use, through the use of webhooks

3 THE INNOVATIVE, MODULAR SMART HOME APP CONCEPT FOR THE BUILDING AUTOMATION

The innovative, modular Smart Home App concept for building automation. The **APP**MODULE brings the innovative, modular Smart Home App concept into building automation. You can mix and match any of the diverse applications that are available to ingrate third-party solutions. With these Smart Home Apps from the dedicated **BAB** APPMARKET, the **APP**MODULE becomes a tailor-made integration unit for your building automation.



Manufacturer of the **APP**MODULE BAB TECHNOLOGIE GmbH

Distribution of all Smart Home Apps for the **APP**MODULE BAB APPMARKET GmbH

Smart Home App developer REAL SMART HOME GmbH

3.1 INFORMATION ABOUT THE APPMODULE

Please refer to the separate product documentation of the **APP**MODULE for a detailed product description and setup instructions.

https://bab-tec.de/appmodule#downloads

Product variants:
The **APP**MODULE is available in three variants:

- **APP**MODULE KNX/TP – for stand-alone use on KNX/TP Bus
- **APP**MODULE EnOcean – for stand-alone use in the EnOcean wireless network
- **APP**MODULE IP – for use in an IP-based KNX installation (KNXnet/IP) or as extension for an EIB**PORT**

4 SMART HOME APP INSTALLATION / UPDATE

Please proceed as follows to install a Smart Home App.

1. Open the **APP**MODULE web page: Enter <IP Address of **APP**MODULE > into your browser's address bar and press Enter. The **APP**MODULE web interface will appear.

2. Log in with your user credentials. Please refer to the **APP**MODULE documentation for login details.

3. Click on the menu entry "App Manager"

4. You are now on the page where already installed Smart Home Apps are listed. The list will be empty if no Smart Home Apps have been installed. Click "Install App" in order to install a new Smart Home App.

5. Now click on "Select App"; a file selector window will appear. Choose the Smart Home App » **IFTTT Connect** « and click "OK".

The Smart Home App » **IFTTT Connect** « must first be downloaded from the **BAB** APPMARKET (www.bab-appmarket.de).

After the message "Installation successful" appears, click "OK". You are ready to configure the Smart Home App.

To update a Smart Home App manually you have to proceed as follows

1. To update an already installed Smart Home App, click on the App icon in the "App Manager".

2. The detail view of the Smart Home App appears. Click on "Update App" to select the Smart Home App package and start the update. The update version must be downloaded from the **BAB** APPMARKET.

After the message "Installation successful" appears, click "OK". The Smart Home App has been updated. Your instance configurations will remain unchanged.

The Smart Home App can also be updated directly in the web interface. Without having to download the Smart Home App from the **BAB**APPMARKET first.
In the “App Manager” available Smart Home App updates are reported

Information
To configurate the Smart Home App please use Google Chrome.

5 SMART HOME APP SETTINGS

With «**IFTTT Connect**» you connect all functions of the **APP**MODULE with the web applications of ifttt.com. Transfer KNX counter readings to Google tables and much more

5.1 IFTTT CONNECT

Note:
After inactivity of 60 minutes the browser session is automatically closed. Unsaved changes will be lost.

As soon as the Smart Home App is installed, you can create so called “Instance”. An Instance is one of several objects of the same class.

In order to create an instance, click on the symbol "Create Instance".

Instance Name:

Choose a name for this new instance.

Comment:

Insert a description what this instance does.

5.2 TELEGRAM TRIGGER FOR IFTTT MAKER EVENT

With the Telegram Trigger you can send actions or events e.g. from the KNX to the IFTTT in order to control functions or services that are connected via the IFTTT.

Trigger Address:

Insert the group address on which to trigger a request to IFTTT.

Data Type:

Insert the data type of the telegrams on the trigger address.

- EIS 1,
- EIS 5: 2 Byte Float
- EIS 6: 0%...100%
- EIS 9: 4 Byte Float
- EIS 10s: 2 Byte -32768...32767
- EIS 10u: 2 Byte 0...65535",
- EIS 11s: 4 Byte -2147483648...2147483647
- EIS 11u: 4 Byte 0...4294967295
- EIS 14s: 1 Byte (-128...127)
- EIS 14u: 1 Byte (0...255)
- EIS 15: Text

Trigger Condition:

Configure which condition must be met in order for a request to be sent to IFTTT.

- Value is equal to threshold
- Value is greater than threshold
- Value is less than threshold
- Value is arbitrary (no threshold)

A note regarding the condition 'is equal to':
Due to possible precision issues, floating point values are considered as equal in this case if their difference is less than 0.01.

Threshold

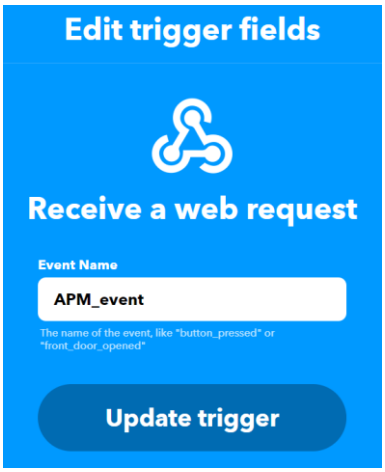
Insert the threshold.

Trigger URL Fixed Component

The fixed component of a an IFTTT trigger URL; here should be entered as the URL
https://maker.ifttt.com/trigger/ .

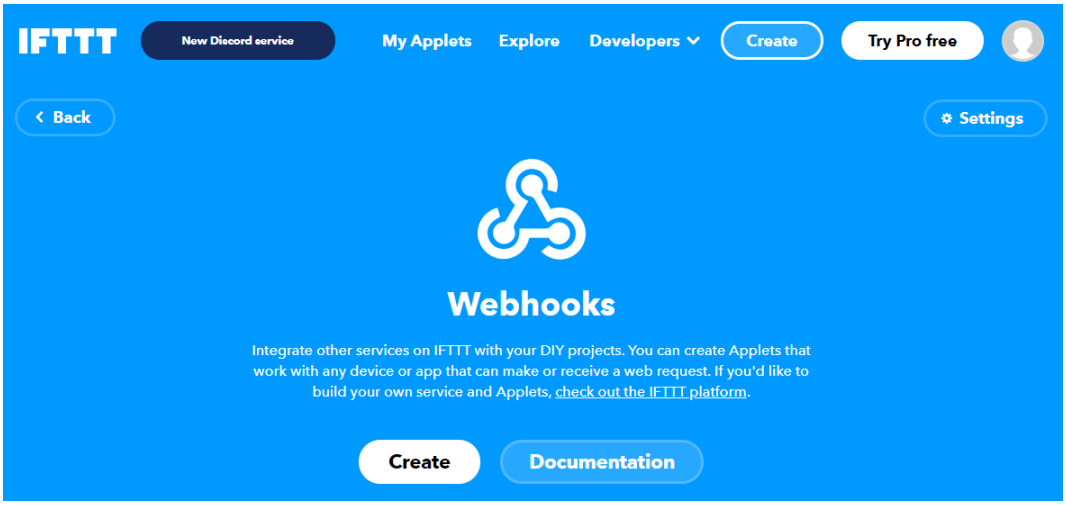
Trigger URL Dynamic Component

The connection to IFTTT is established via the service „Webhooks". In this respect an IFTTT applet is created with „Webhooks" as the „IF"-condition of the applet and the function to be carried out as the „Then" action.
The dynamic part pf the URL has the following format:
{Event}/with/key/{YOUR_KEY}



For {Event} insert the event name in the Webhooks editor:

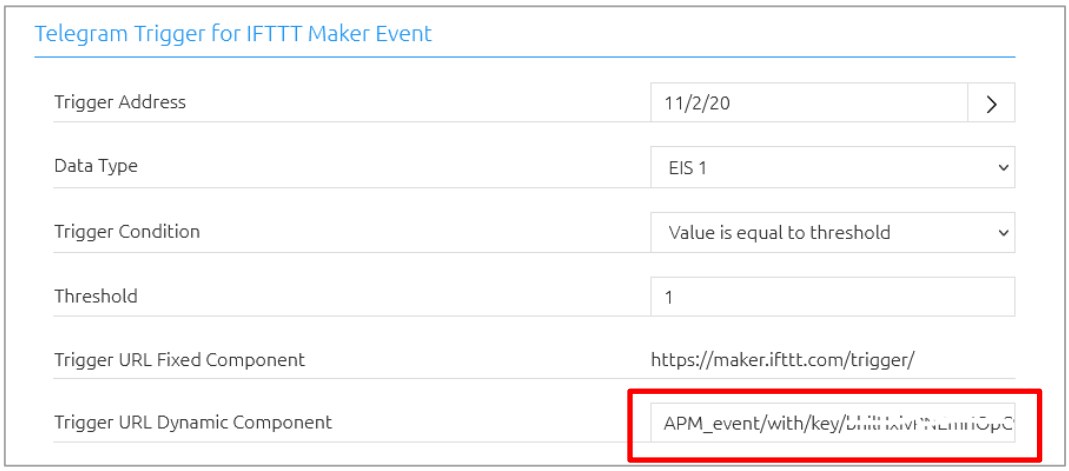
You can find your KEY for „Webhooks" in your IFTTT account under „My services" in the menu item „Webhooks".



Click at „Webhooks" the button Documentation, to copy your IFTTT Key into the **APP**MODULE.

,

You can use it to complete the trigger URL.



HTTP-status code (EIS 10 0–65535)

Enter the group address to which the respective HTTP status code of the IFTTT server should be sent. If the web request was successful, it will be sent a code 200. Missing or incorrect token data usually results in code 401. You can find more information on http status codes on the Internet.

5.3 VALUES TO INCLUDE (OPTIONAL)

"Linked values" can be used to control actions or events by the WEB, linked functions and services of the IFTTT.

Values Addresses:

A list of group addresses used to exchange values between this instance and IFTTT. You can either send values from these group addresses to IFTTT, receive values from IFTTT via web request which will then be sent to the corresponding group addresses as a telegram or do both. Data from IFTTT to this instance must be sent as JSON. You can view the latest request from IFTTT to this instance via link in order to check the validity of the data sent from IFTTT.

Value Address Key:

Assign a key to the value group address. This key is sent along with the value to IFTTT and it also serves to send a value received from IFTTT to the correct group address.

Group Address:

Insert the group address for the value.

Data Type:

Insert the data type of the telegrams on this value address.

- EIS 1,
- EIS 5: 2 Byte Float
- EIS 6: 0%...100%
- EIS 9: 4 Byte Float
- EIS 10s: 2 Byte -32768...32767
- EIS 10u: 2 Byte 0...65535",
- EIS 11s: 4 Byte -2147483648...2147483647
- EIS 11u: 4 Byte 0...4294967295
- EIS 14s: 1 Byte (-128...127)
- EIS 14u: 1 Byte (0...255)
- EIS 15: Text

5.4 EXTERNAL ADDRESS OF THE APPMODULE (FOR WEBHOOKS)

Internet Address:

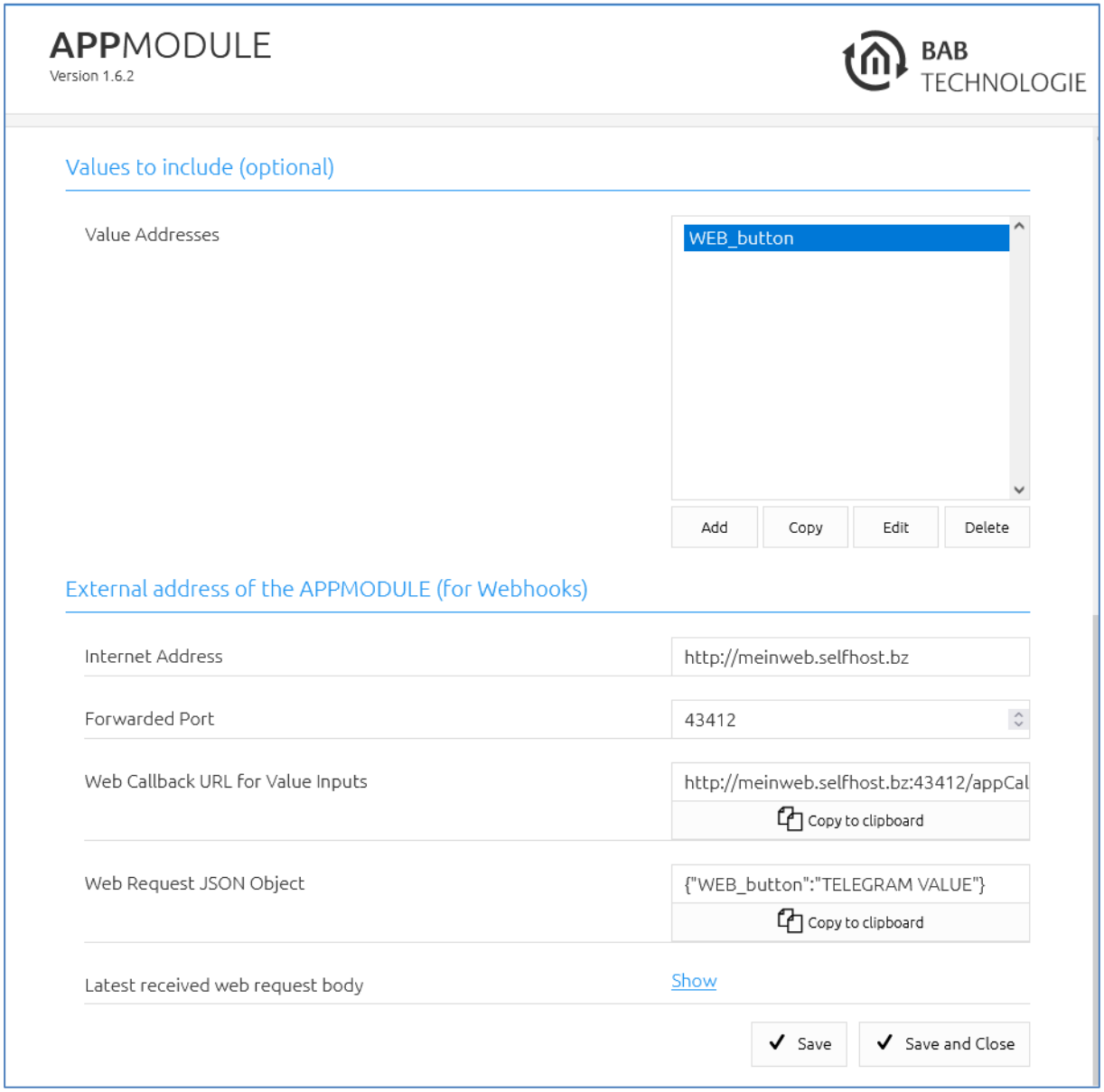
Insert your external internet addresses (DNS; DynDNS or public IP address). Please make sure to include the protocol (“http://” or “https://”).

Forwarded Port:

Insert the port number which you configured as port forward to port 81 (http) or port 444 (https) of the **APP**MODULE.
(Example port 81 to 43412 or if port 444 to 43422)

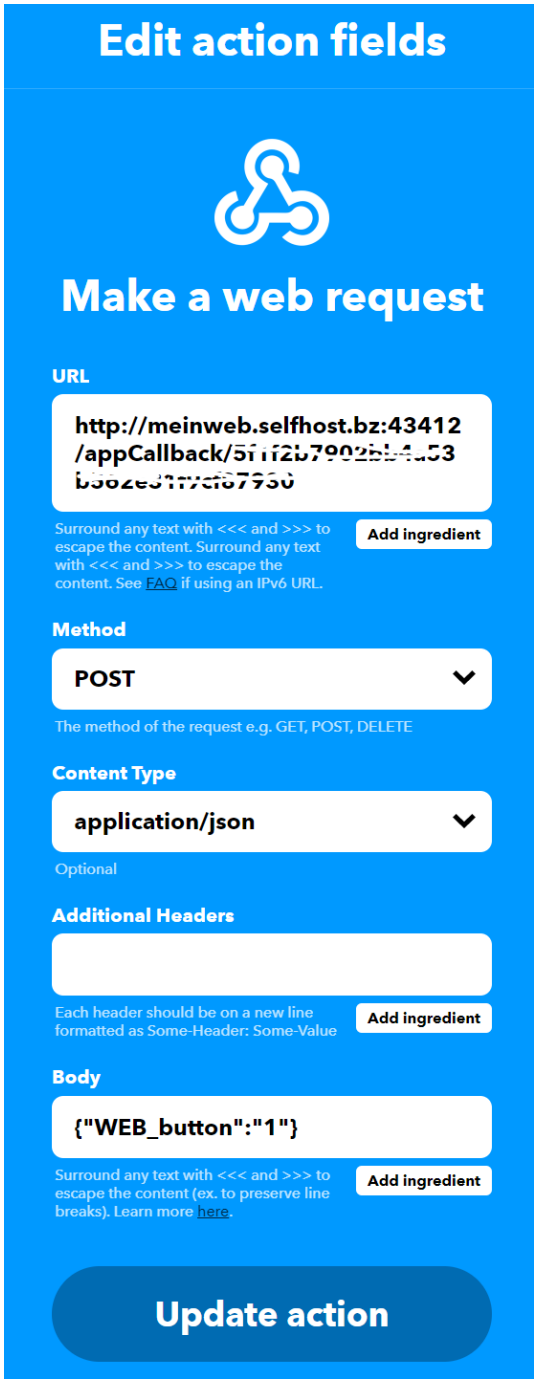
Web Callback URL for Value Inputs:

This field displays the generated Web Callback URL (Webhook) for this instance which contains both the internet address and forwarded port. Click on the button below to copy the address to the clipboard. Enter this URL into the URL field at IFTTT when creating »Action« (Webhooks).


APPMODULE
Version 1.6.2
BAB TECHNOLOGIE
Values to include (optional)
Value Addresses
WEB_button
Add
Copy
Edit
Delete
External address of the APPMODULE (for Webhooks)
Internet Address
http://meinweb.selfhost.bz
Forwarded Port
43412
Web Callback URL for Value Inputs
http://meinweb.selfhost.bz:43412/appCal
Copy to clipboard
Web Request JSON Object
{"WEB_button":"TELEGRAM VALUE"}
Copy to clipboard
Latest received web request body
Show
Save
Save and Close

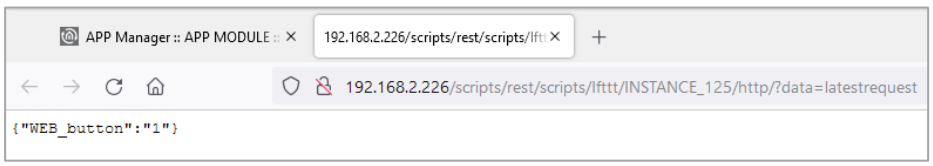
Web Request JSON Object:

This field displays a template of a JSON object which includes a property for each configured value address. Click on the button below to copy the JSON object to the clipboard and simply replace the placeholder “TELEGRAM VALUE” with a desired value that you want to be sent to the corresponding value address via an IFTTT web request. Remove a property altogether if you do not wish a value to be sent to a certain value address. Insert the modified JSON object onto the Body field when creating an Action (Webhooks).



Latest received web request body:

Here you can display the last Web request body received.



6 APPLICATION EXAMPLE

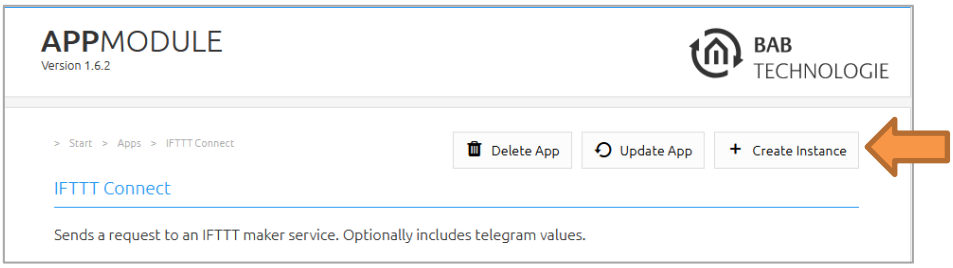
Important note: The application examples were tested at the time of creation. Due to software changes by third parties, it can happen that full functionality is no longer guaranteed. In this respect, use these examples only as suggestions for your own applications. The examples are intended for specialists who have the necessary technical knowledge.
A guarantee on the function is therefore excluded.

6.1 INSTRUCTION: LOCATION (GEOFENCE) TRIGGER KNX GROUP ADDRESS

These "step by step" instructions show the configuration of the IFTTT Connect app to trigger location-dependent actions (geofence) with a KNX telegram.

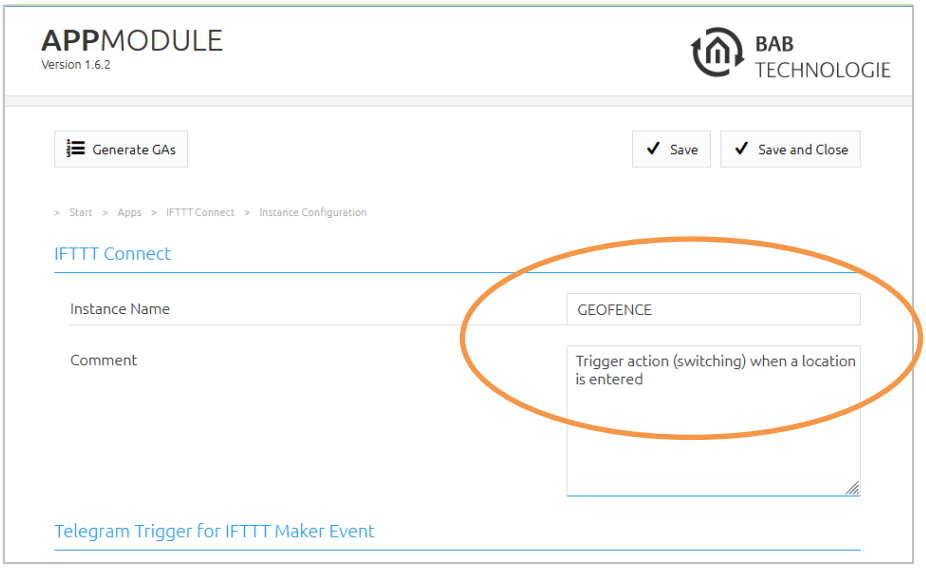
CONFIGURATION IN **APP**MODULE

First, a new instance is created in the **APP**MODULE.



Create instance

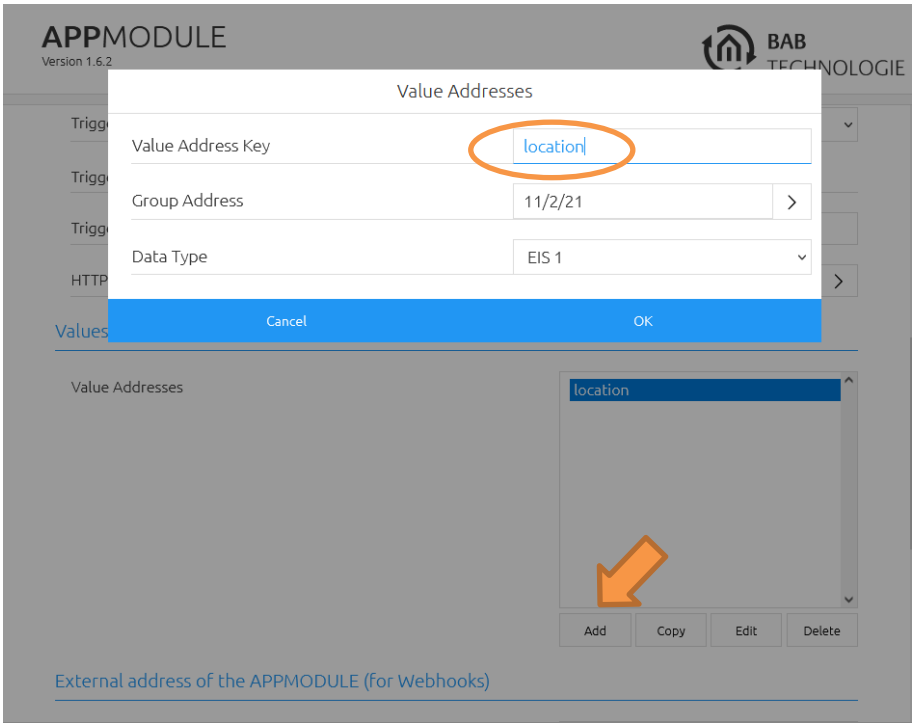
Next, enter name of the instance and a make your comments describing this instance.



The next step is to create **Value Address**. This KNX group address is used when an applet is triggered via IFTTT. In this example, this happens when entering a geographically defined area.

In the figure is used the 1-bit KNX group address **11/2/21**.

The registered name **location** is the identifier that must be specified later in the applet.

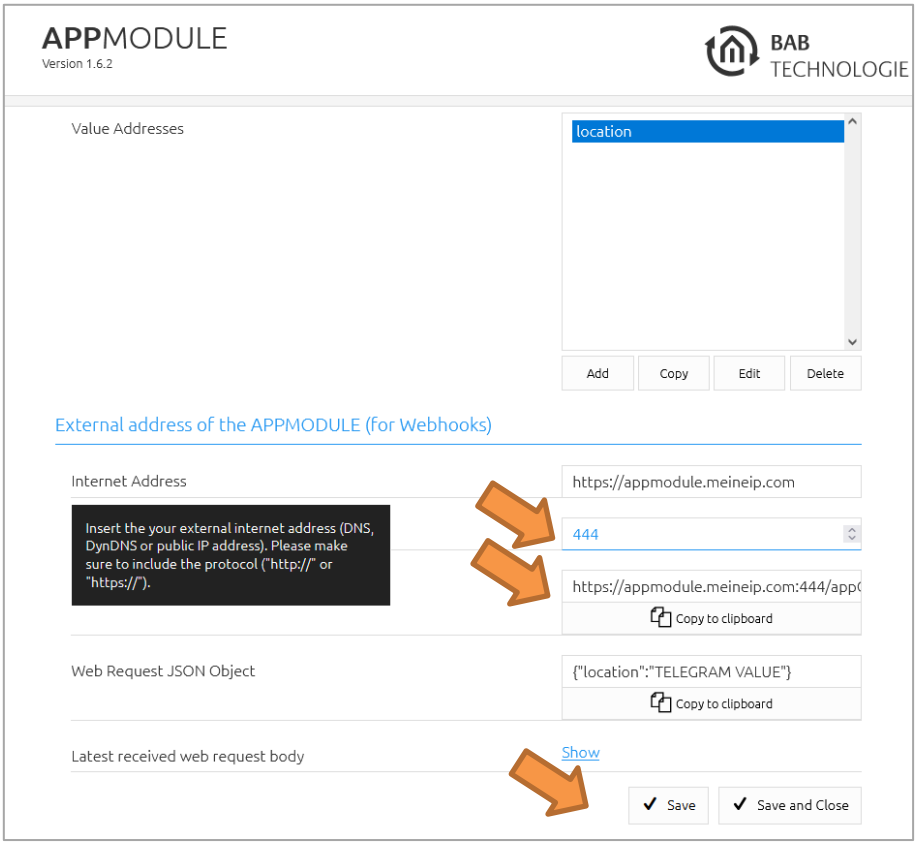


After the Value Address the complete **Internet address** and the **forwarded port** are entered.

At field **Internet address** the external Internet address, under which address the **APP**MODDULE can be reached for the IFTTT service must be entered (can be e.g. a host name by dyndns address or public IP address).

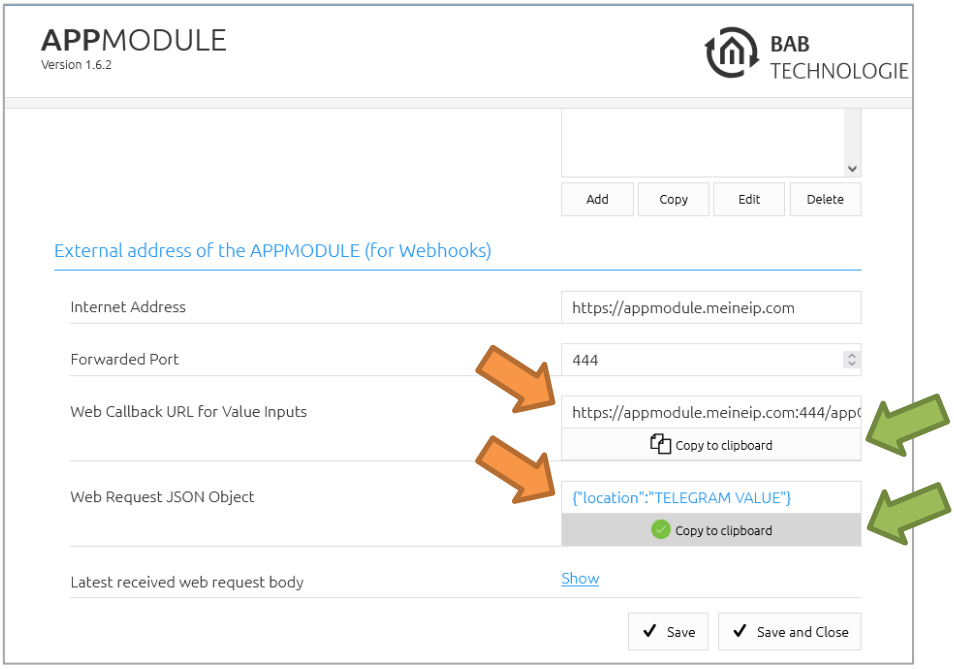
Under Forwarded Port is entered the port that is released in the Internet router and forwarded to the **APP**MODDULE.

The **APP**MODDULE allows only port 81 (for http) or port 444 (for https) to be used.



Only after saving the instance does the IFTT Connect APP automatically generate the **URL** for the **web request** and **web request JSON object** that are required for the IFTTT request.

After saving and reopening the instance, the required information is displayed (orange) and can be copied (green) simply to enter it in the IFTTT recipe.



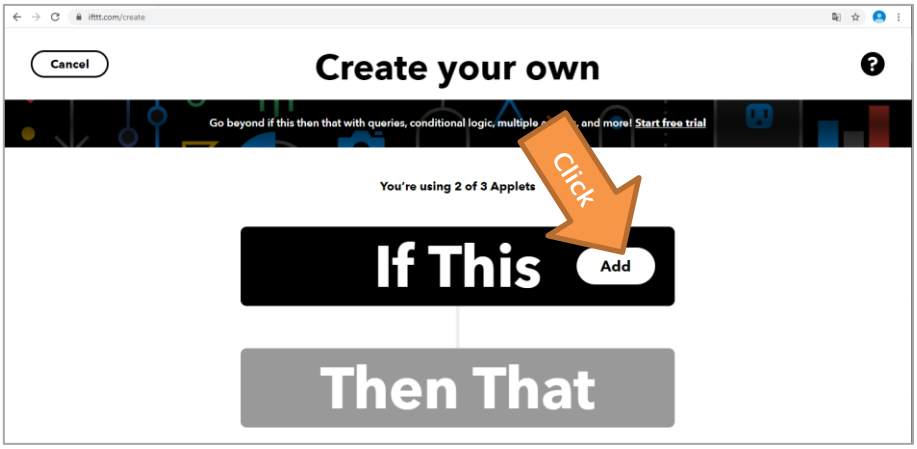
The URL for the web request is generated and given an ID that can be assigned to this instance. Here is an example, how the URL is structured (x stands for alphanumeric characters and is generated by the **APP**MODULE):
appmodule.meineip.com/appCallback/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The Web Request JSON object is also generated automatically and uses the previously created value address. The placeholder "**TELEGRAM VALUE**" is replaced by the user in the IFTTT applet with the desired telegram value! {**"location":"TELEGRAM VALUE"**}

This completes the preparation on the **APP**MODULE side. The configuration page of the instance is still required and can therefore remain open.

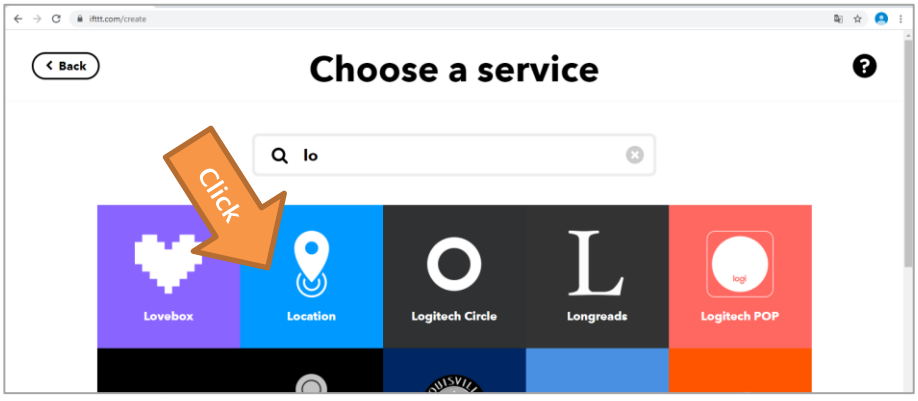
CONFIGURATION IN IFTTT WEBINTERFACE

After logging in to www.ifttt.com at first step it is to create a new applet. The new applet is accompanied by an assistant. The assistant is started under **Create**.

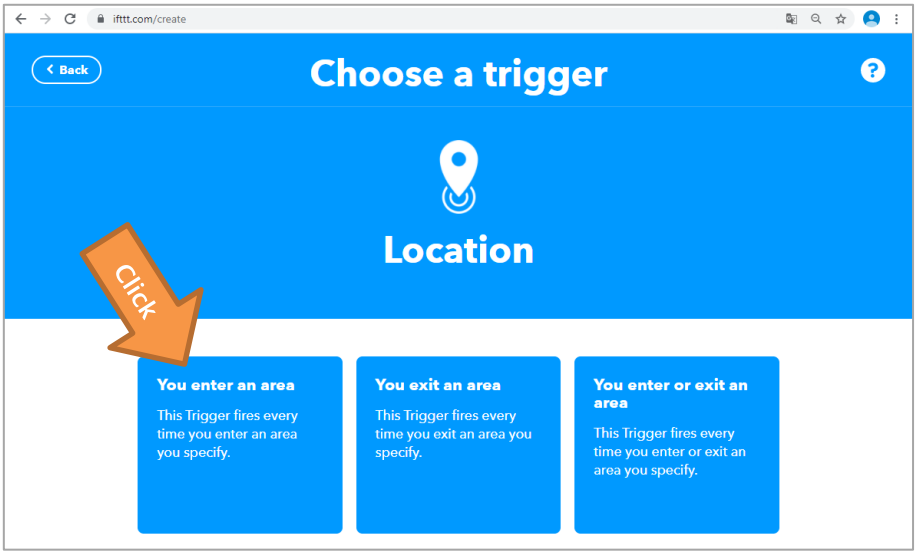
The first step is to define **This**.

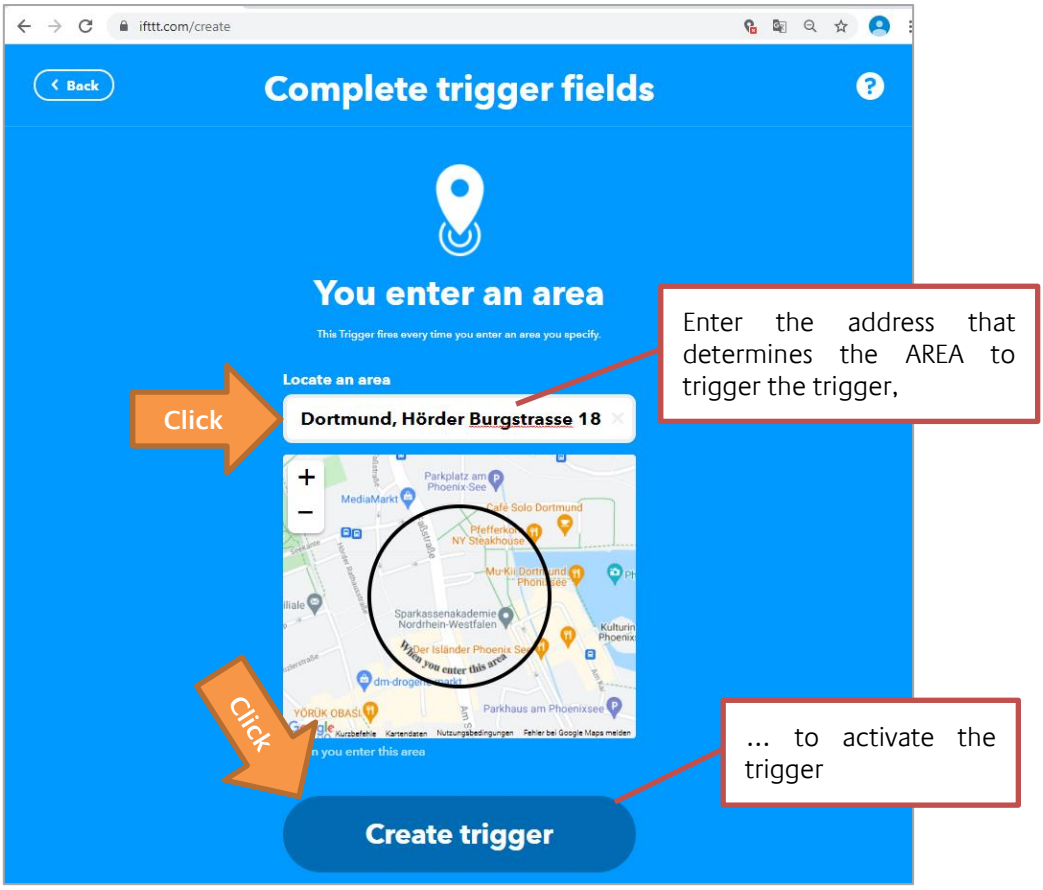


For this purpose, the **Location** service is selected as the trigger.

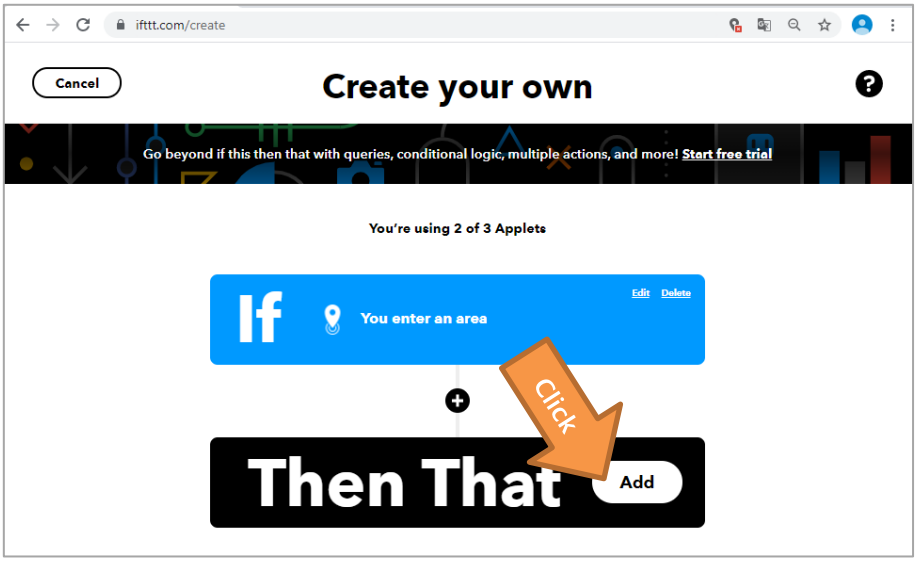


In this example the trigger will be "**You enter an area**". When you register with your mobile device and the APP installed on it that you have reached the AREA, the trigger is caused.

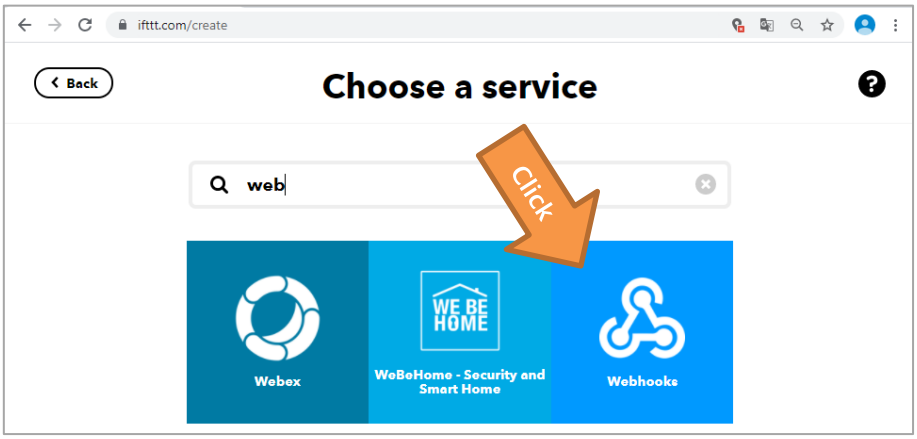




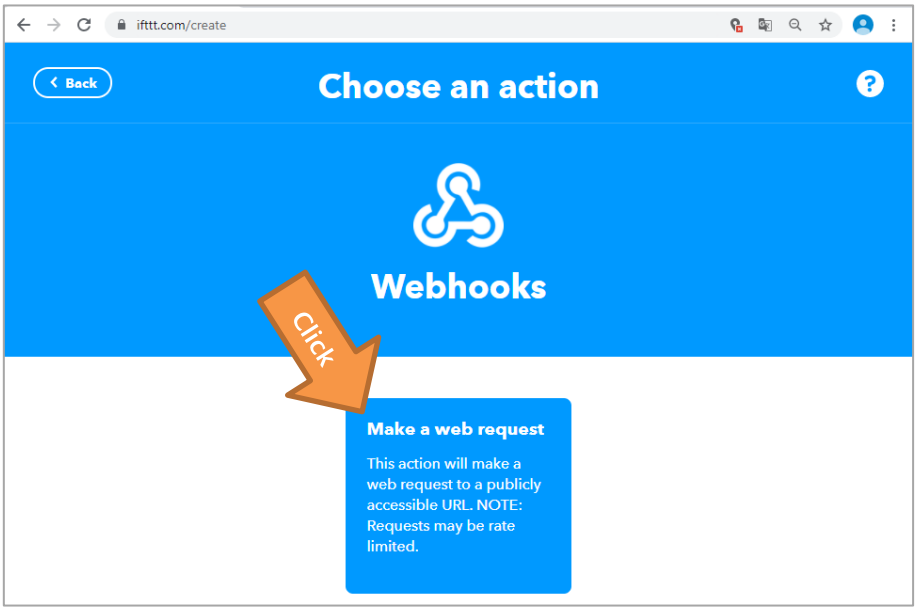
After the **This**, the **That** is selected and configured.



To address the **APP**MODULE via IFTTT, the **Webhooks** service is required



An action is available for Webhooks. "**Make a web request**" is the desired function.

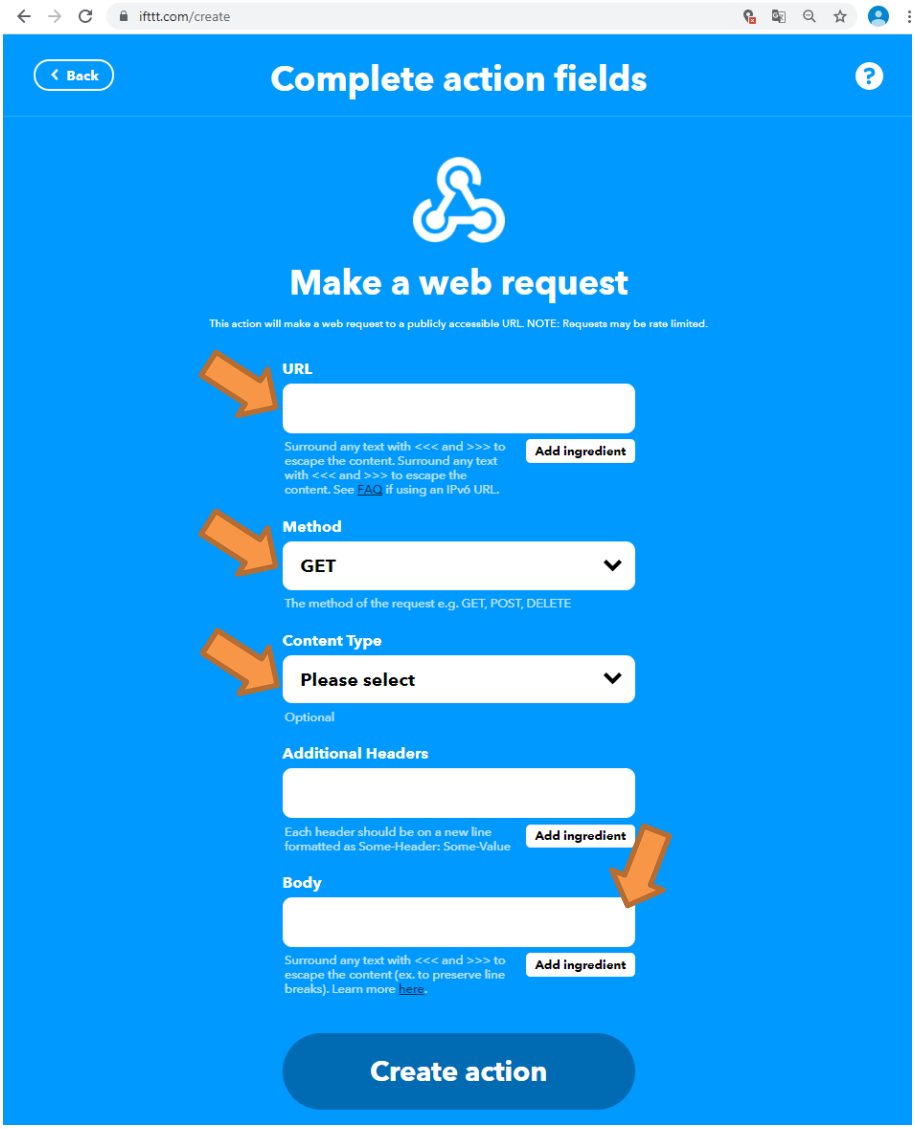


The penultimate step is the **URL**, the address at which the **APP**MODULE can be reached, and the data required, which information is transferred to the **APP**MODULE.

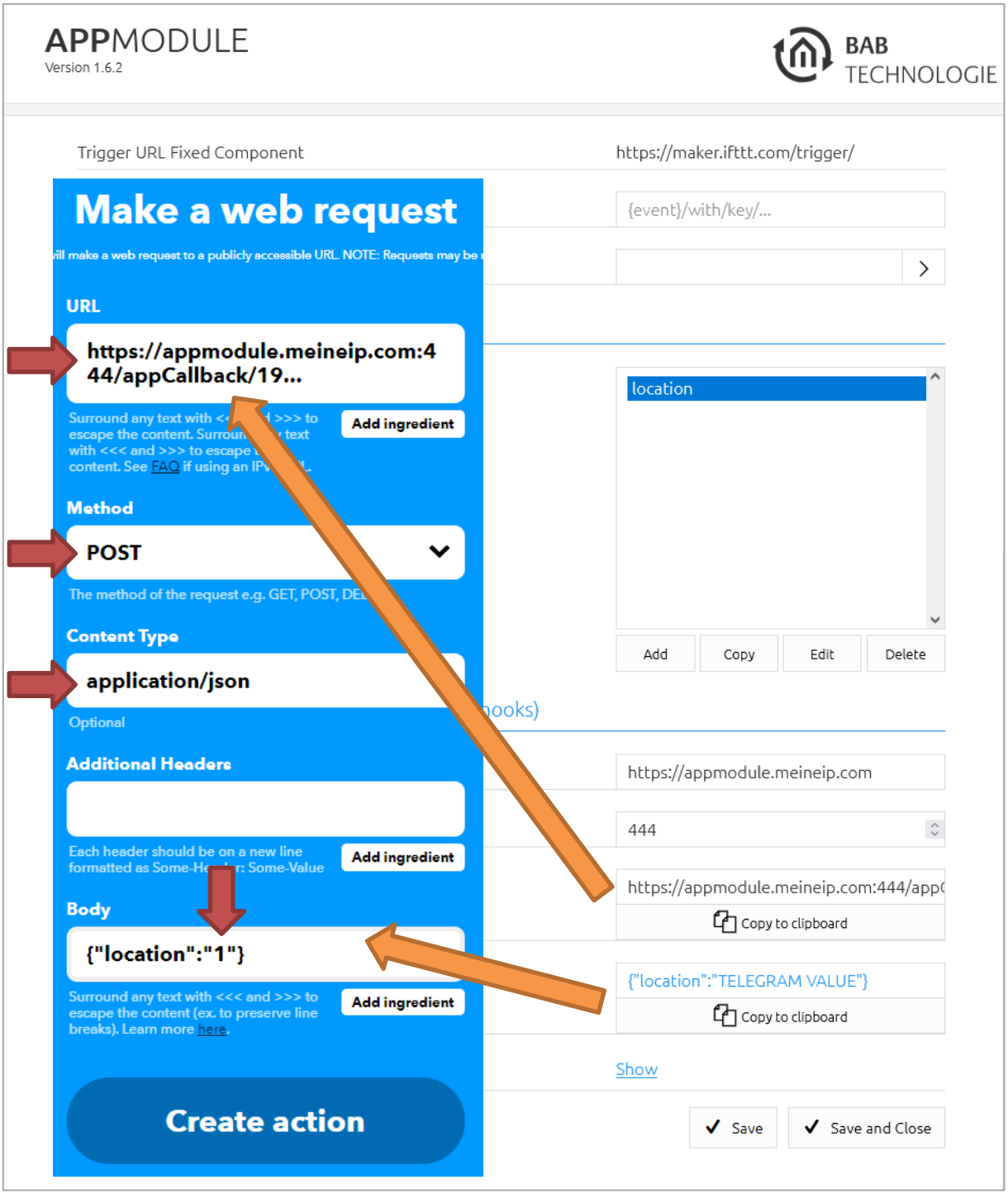
The content for the **URL** and **body** can be copied and pasted from the web interface of the **APP**MODULE using the copy function.

NOTE1: The protocol must be added in the URL field! So http:// if port 81 or https:// if port 444 was forwarded.

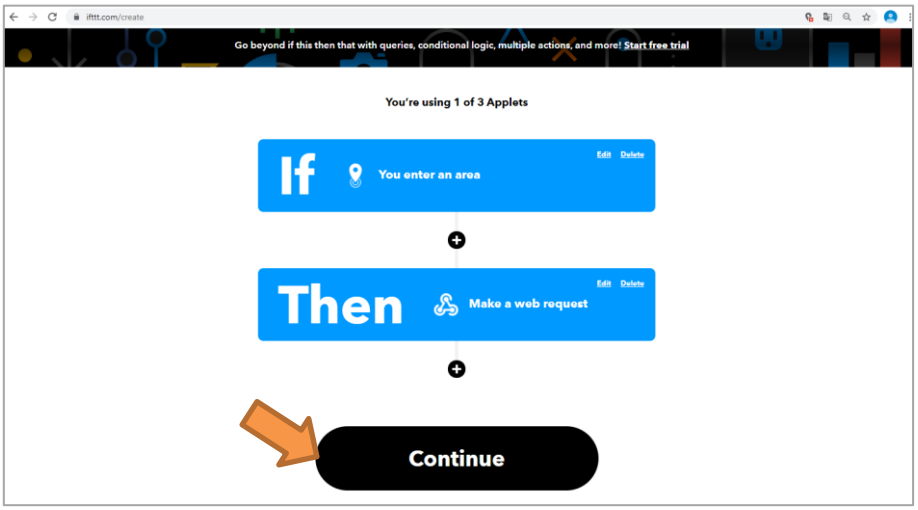
NOTE2: The placeholder **TELEGRAM VALUE** in the body must be replaced by the desired telegram value. In this example, the value "**1**" is used. This means that each time a mail is received, the value 1 is sent to the KNX group address entered in the **APP**MODULE.

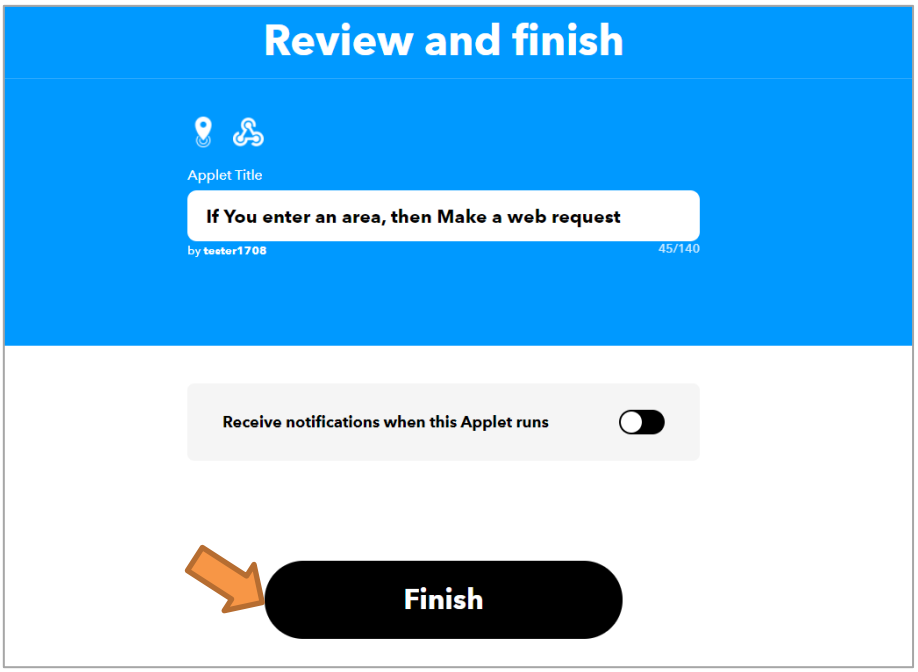


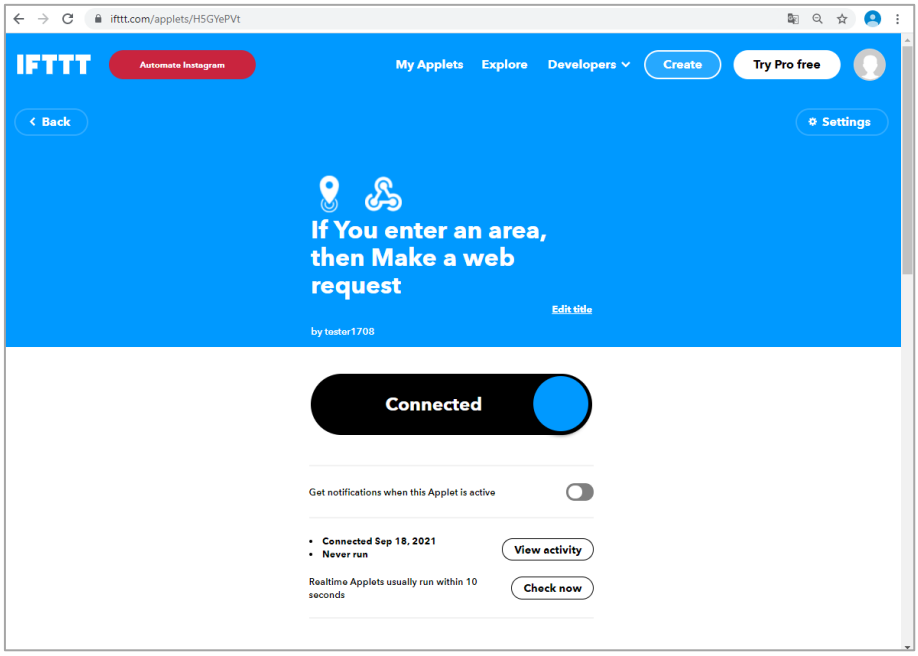
Here is an overview of the web interface of the **APP**MODULE and the configuration of the IFTTT applet.



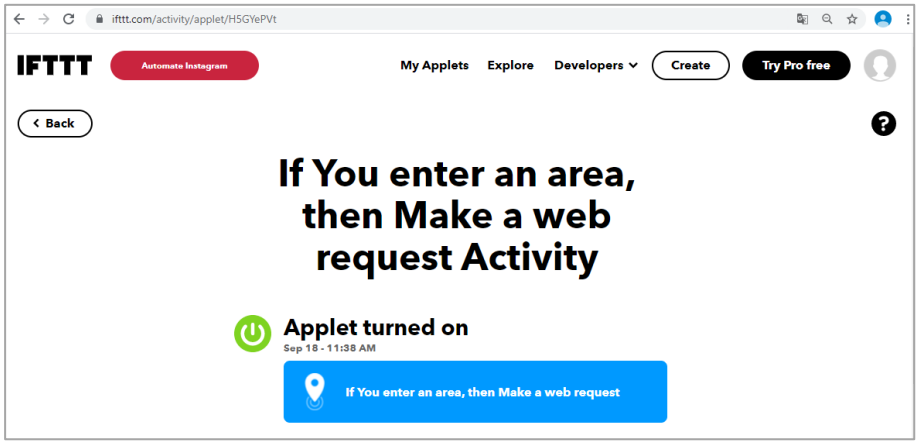
The last step is to give the applet a description and to close the wizard with Continue and **Finish**.

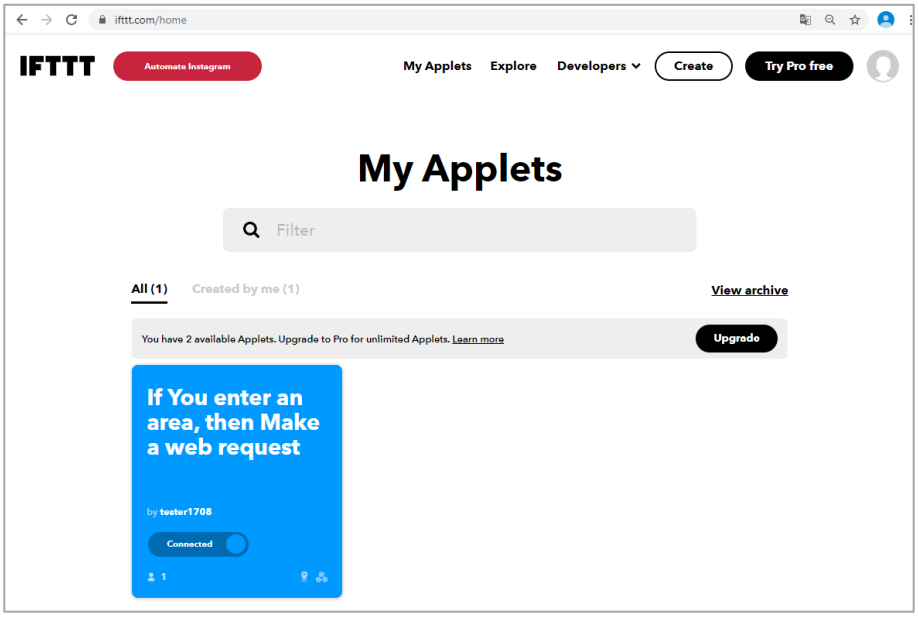






The new applet is automatically listed under the "My Applets" menu and can be edited if necessary. (e.g. change of location)

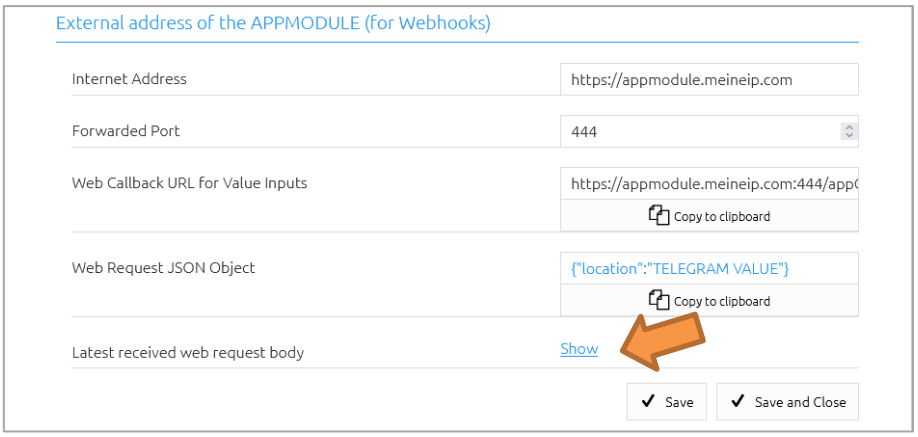


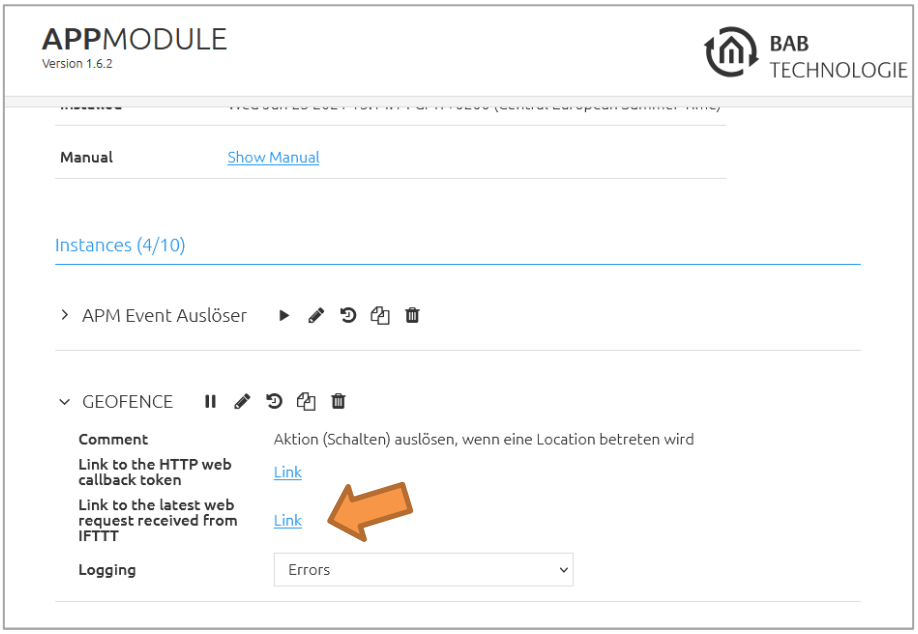


This completes the preparation on the IFTTT side and the configuration can now be tested.

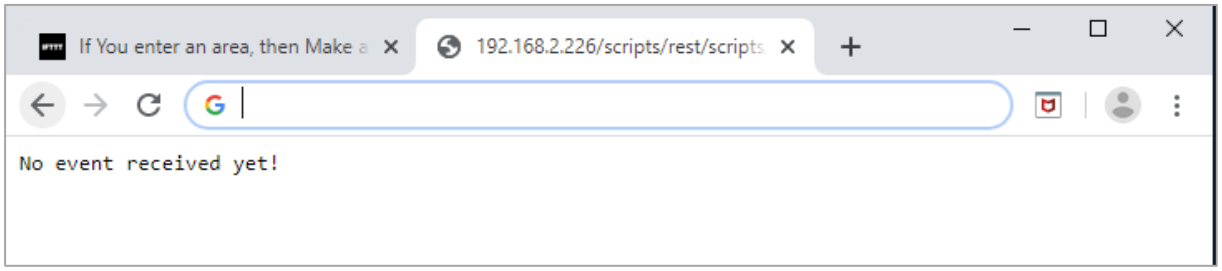
TESTING THE FUNCTION

When you click on Show, as well as in the control instance, there is a link (orange) with which the last received web request from IFTTT can be called up. This can be used to check the function.

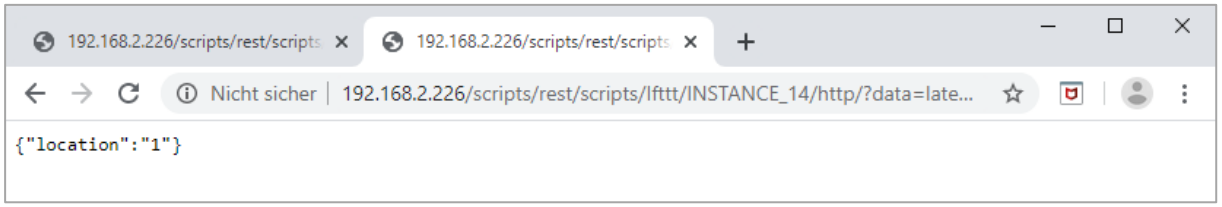




The first call of this link (before a mail is received!) Is the correct output **„No event received yet!“**.



Before this link is called a second time, the IFTTT applet must first be triggered. In order to trigger the applet, the location function has to work. The way the location function works depends on the IFTTT-APP settings of your mobile device and of course the location you have set.
When entering the area of your location, the applet is triggered and when this page is reloaded, the output changes and reports the stored JSON data "**{"location":"1"}**".



The reception from the IFTTT applet to the **APP**MODULE was successful.
The group address entered in the **APP**MODULE starts the associated scene when it is received, depending on your mobile device.

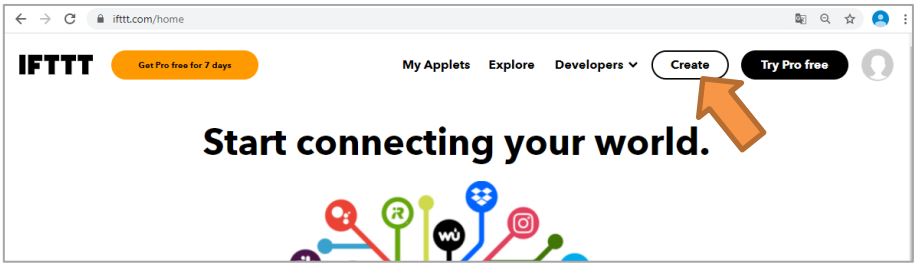
6.2 INSTRUCTION: TELEGRAM VALUE FROM KNX TO GOOGLE DRIVE STORAGE

These "step by step" instructions show the configuration to save telegram values together with a time stamp in a table on Google Drive.

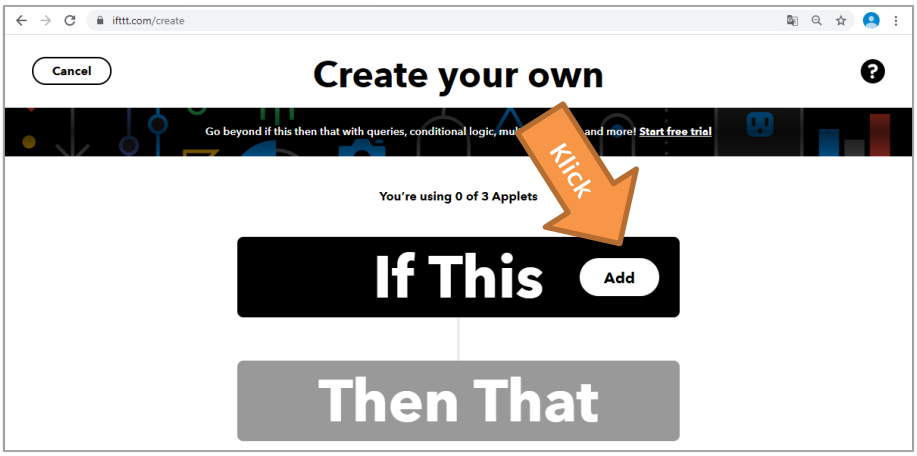
Important note: The application examples were tested at the time of creation. Due to software changes by third parties, it can happen that full functionality is no longer guaranteed. In this respect, use these examples only as suggestions for your own applications. The examples are intended for specialists who have the necessary technical knowledge.
A guarantee on the function is therefore excluded.

CONFIGURATION AT IFTTT WEBINTERFACE

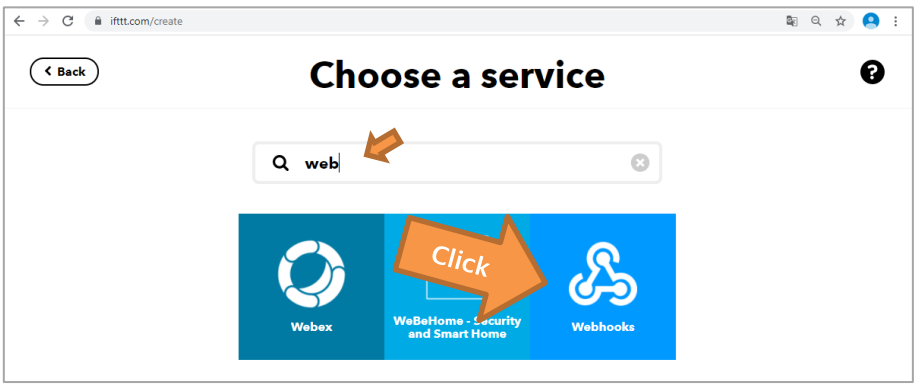
After logging in to www.ifttt.com at first action to create a new applet. The new applet is accompanied by an assistant. The assistant is started under My Applets / Create (orange).



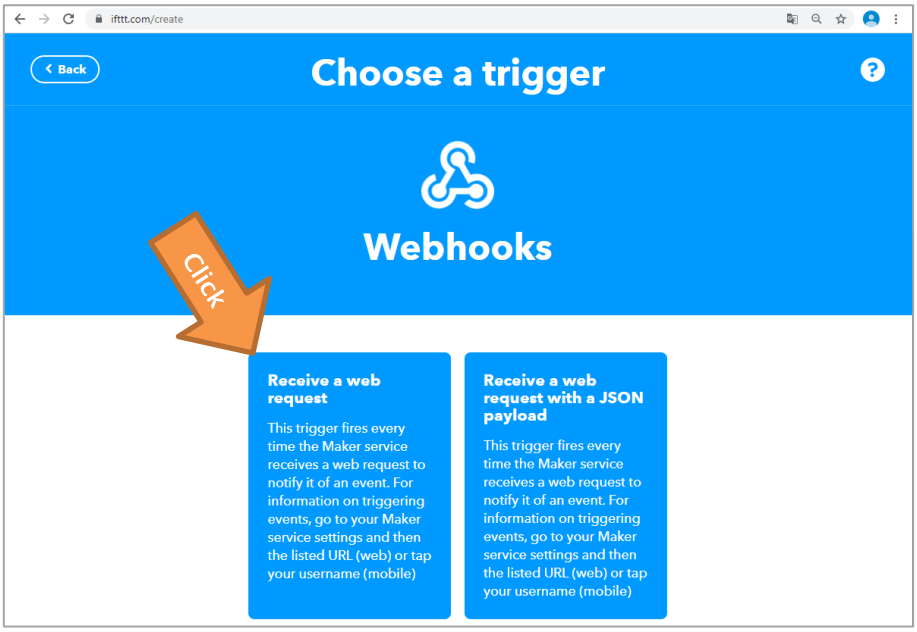
The first step is to define the **This**.

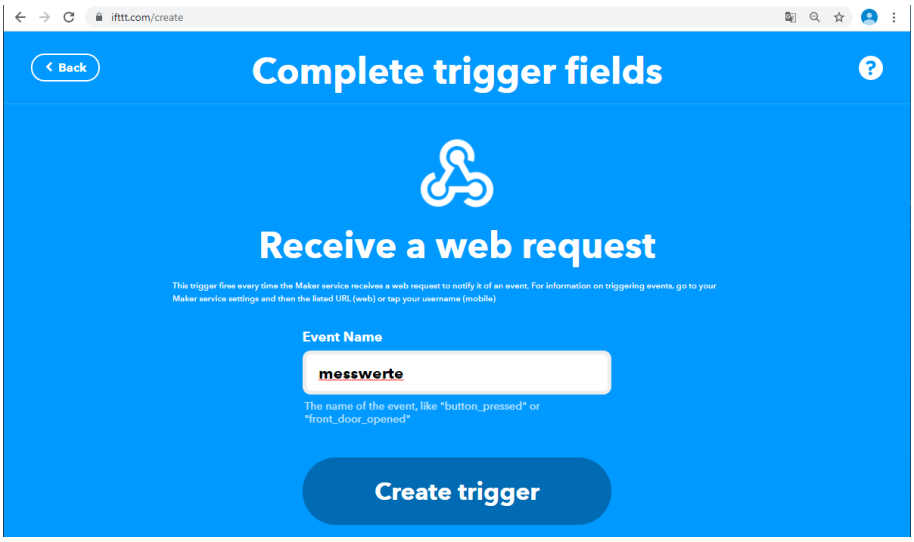


In this example, the values are sent from the **APP**MODULE to the IFTTT service via http request. Thus, the http request can be used as a trigger.
The required service for this is the service "Webhooks".



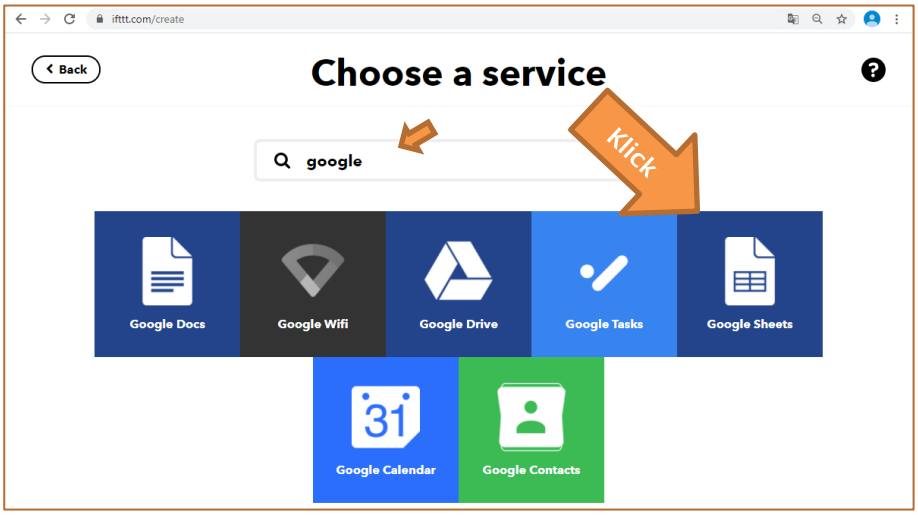
The trigger "**Receive a web request**" (orange) is available in this service. This trigger is triggered as soon as an http request (from the **APP**MODULE) is received.





Enter a name of the event: "messwerte"

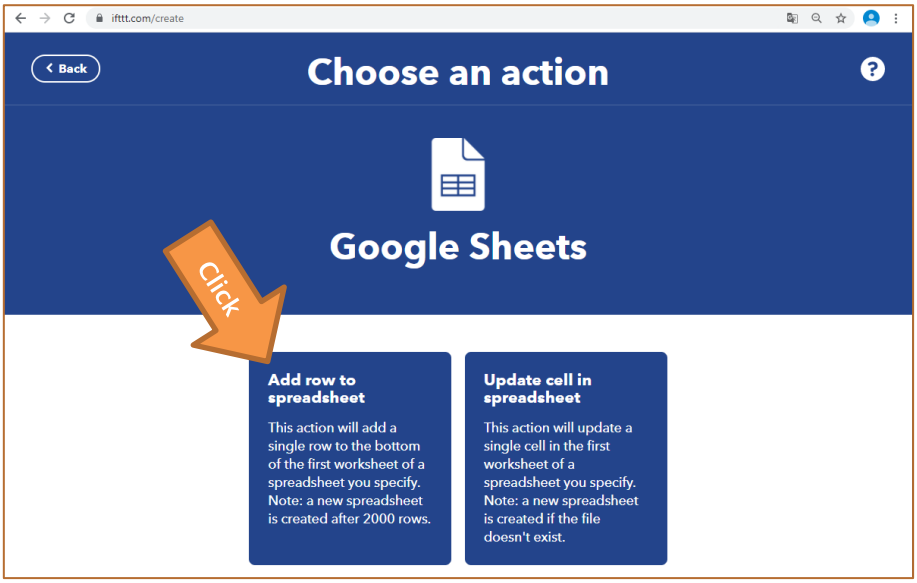
After the This, the **That** is selected and configured.



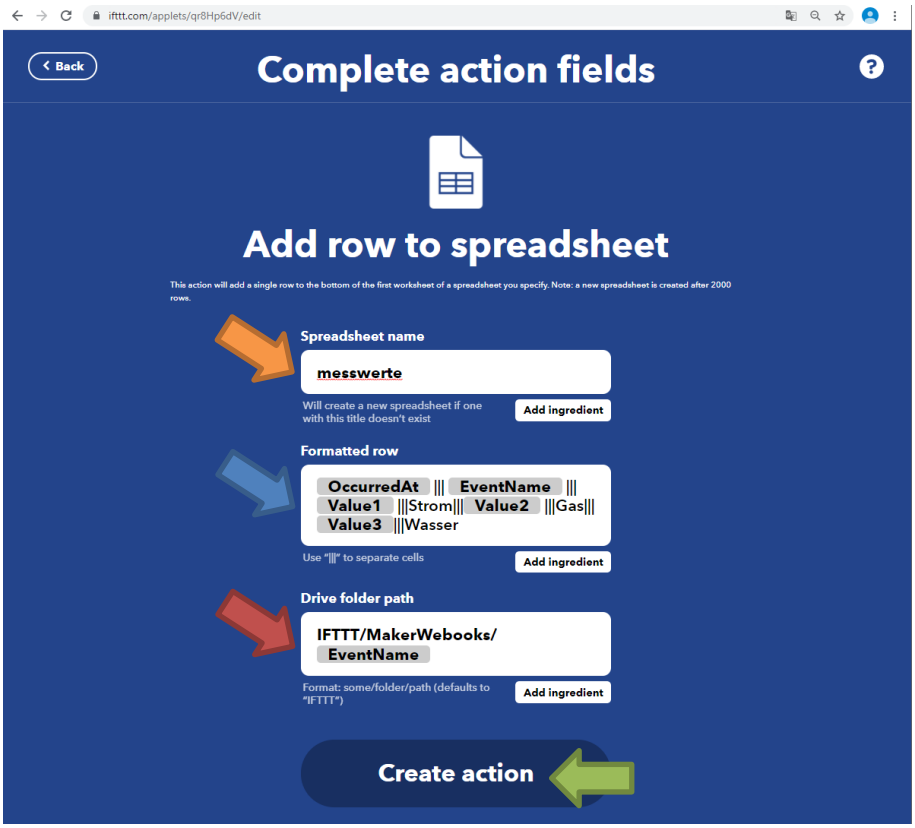
If the IFTTT service is addressed by the **APP**MODULE via http request, the desired action (action service) is a function in the web-based spreadsheet software from Google (Google Sheets).

The desired function in the spreadsheet software is to add the values transferred by the **APP**MODULE to a new line of the selected table.

"**Add row to spreadsheet**" is the desired action.



The penultimate step is the configuration of the table.



The name of the table is entered here first. The table is created on the Google Drive under this name. In this example, the table is called "messwerte" (orange). (messwerte = measured values)

Then the formatting (blue) of the table row is defined. The string "|||" stands for a column change. The following formatting was used for this example.

```
{{OccurredAt}} ||| {{EventName}}
||| {{Value1}} ||| Strom ||| {{Value2}} ||| Gas ||| {{Value3}} ||| Wasser
```

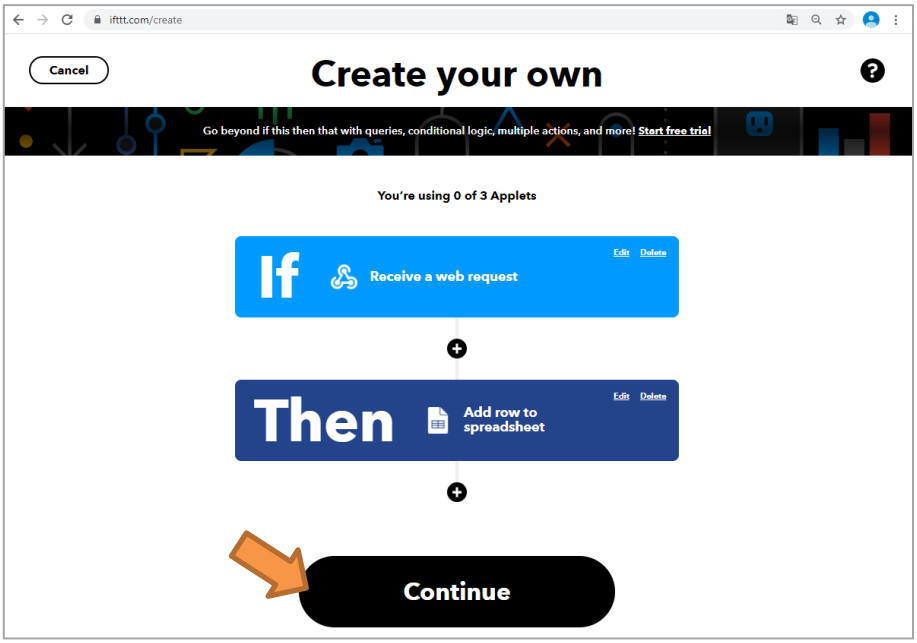
This means that the rows in the table contain the following columns:

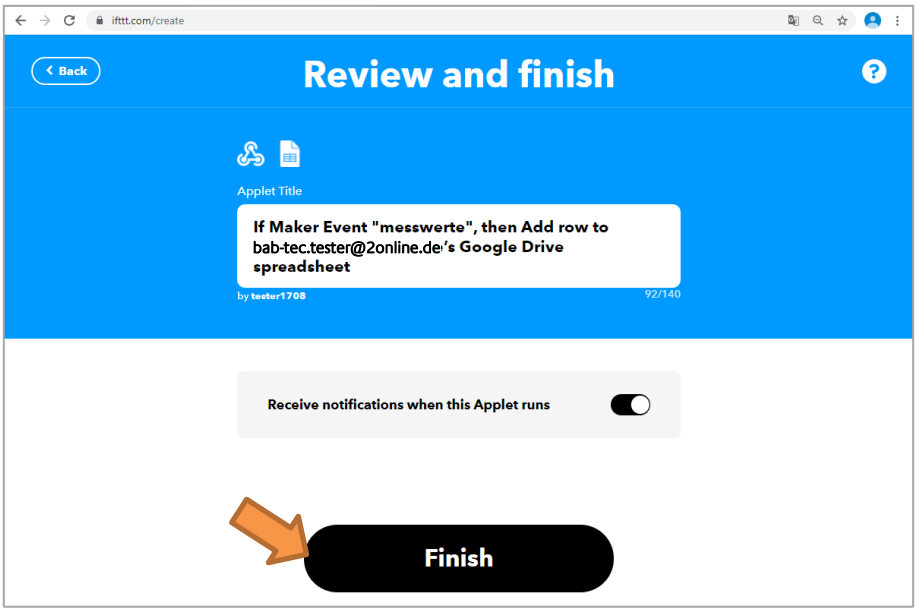
Time stamp; Name of the applet; 1st value; Text "Strom"; 2nd value; Text "Gas"; 3rd value; Text "Wasser"

The last option is the storage location of the table on Google Drive. This means the directory path of the table.

In this example, the table is saved in the "messwerte" directory under the path "IFTTT / MakerWebooks /".

The last step is to save the applet via “Continue” with “Finish”.



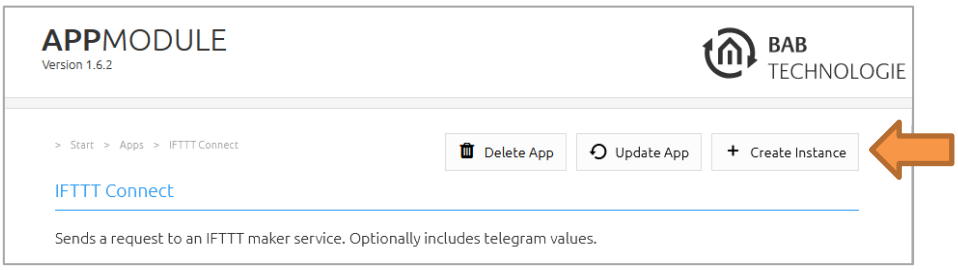


This completes the preparation on the IFTTT side. Next, the IFTTT instance is created on the **APP**MODULE according to the configuration made.

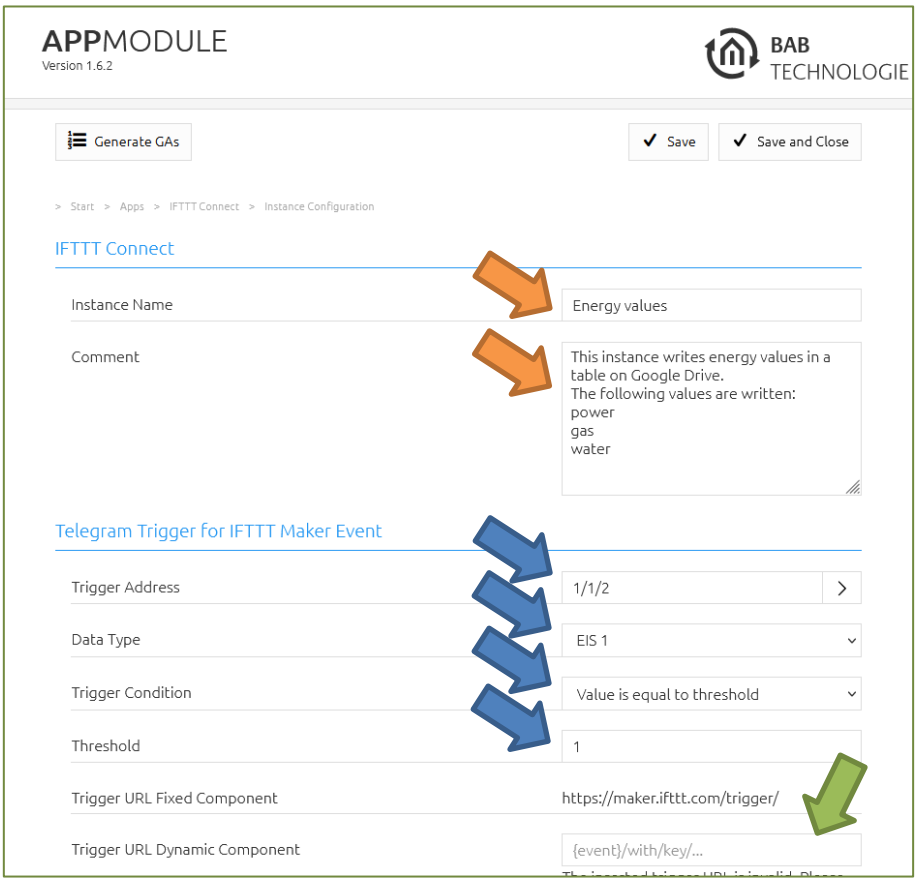
CONFIGURATION **APP**MODULE

The “IFTTT Connect” smart home app is called up in the App Manager in the **APP**MODULE web interface.

The first step is to create an instance under the “IFTTT Connect” smart home app (orange).



A name and description must be entered for the instance (orange).



The next step is to select the trigger address (blue). The data type and the trigger condition (blue) are expected for the specified trigger address.

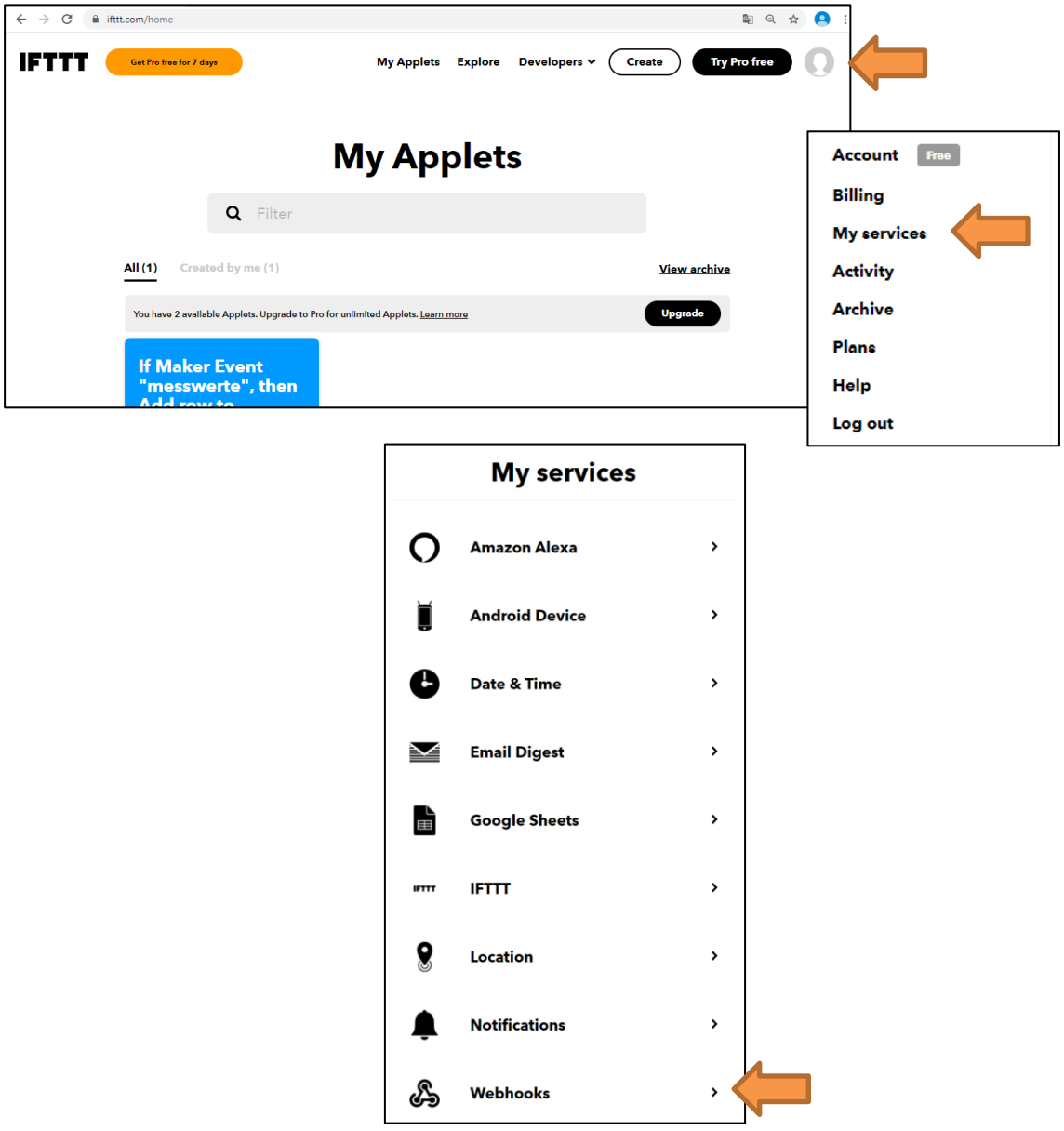
If the condition is not “equal” a threshold value is also necessary.

In this example, the instance is triggered when a “1” is received via the KNX group address "1/1/2".

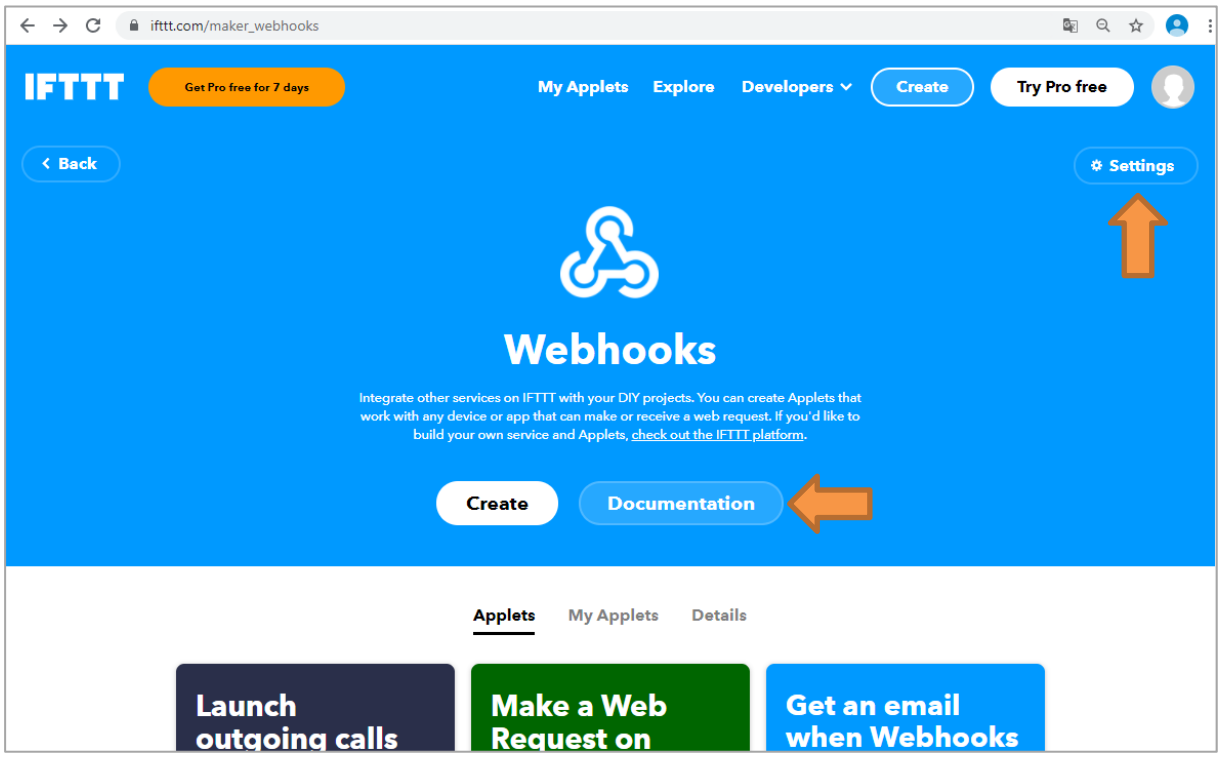
Next, the “trigger URL” is specified. This requires a key so that the request can be assigned to the correct account.

The key can be found on the IFTTT web interface under "My Applets / Service / Webhooks / Settings".

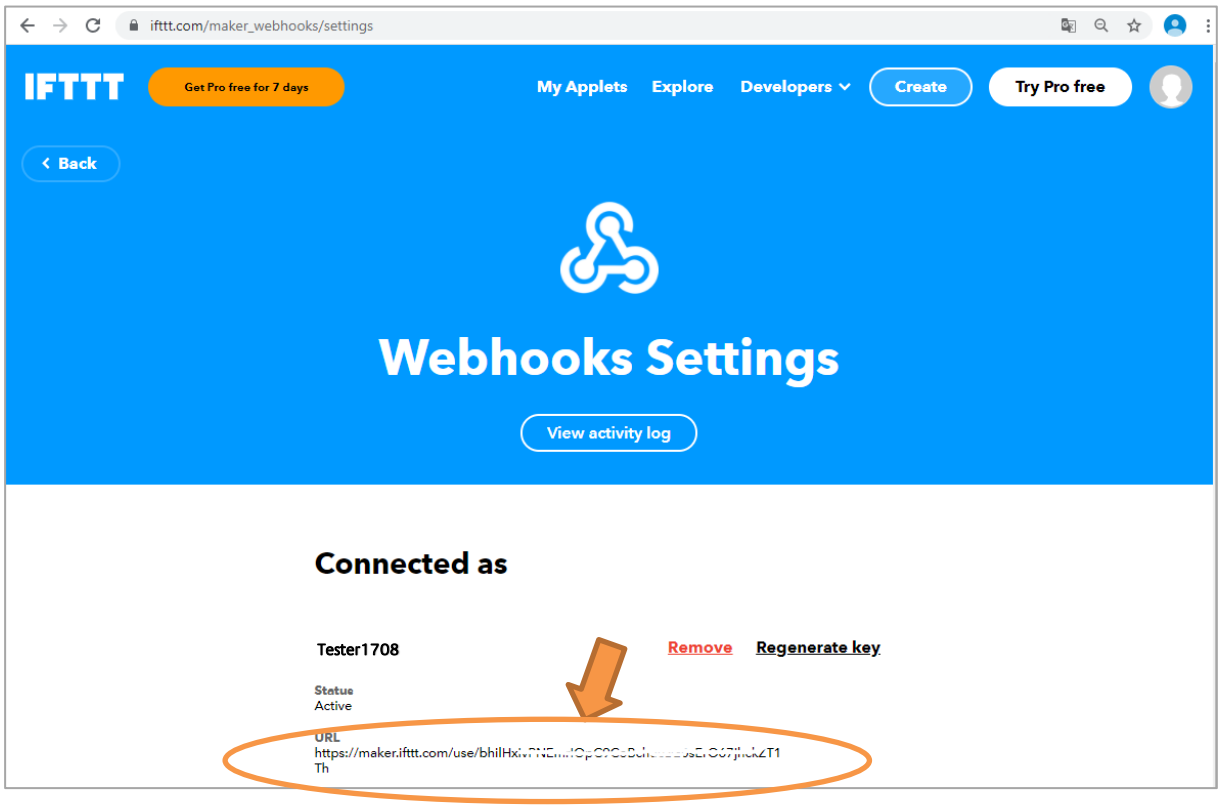
To get to these settings, click on your IFTTT account, go to "My services" and select the Webhooks service.



The configuration (key) of the "Webhooks" service is available under "Settings" (orange) or via the “Documentation”.



The required key is a 43-digit combination of letters, numbers and characters. This must be copied and stored in the instance in the "Trigger URL" field.



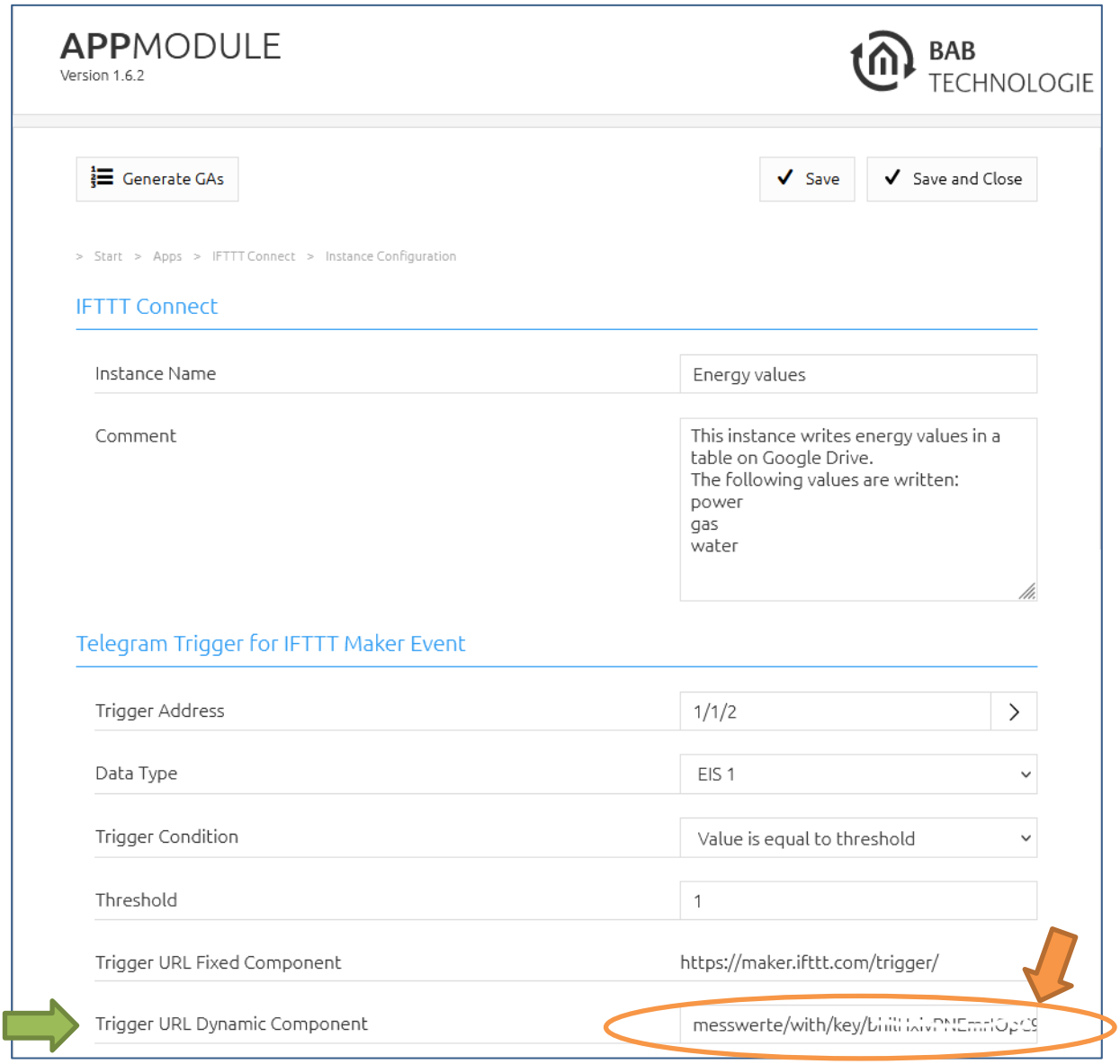
The trigger URL (green) is always structured as follows:

{event}/with/key/{key}

„{event}“ is the assigned event name that was assigned when the applet was created. The middle part **"/ with / key /"** is always the same and must be used as it is. And the **"{Key}"** is copied from the setting of the webhook service (orange).

This results in the following trigger URL for the example:

messwerte/with/key/XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



Finally, only the value addresses are missing via which the values to be recorded are received via the KNX bus. These are created in the "Value addresses" list element.

The "Add" button (orange) opens a dialog window in which the parameters can be entered. The parameters are **Value Address Key, Group Address** and **Data Type**.

The same identifier (lower case!) As it appears in the applet configuration must be used for the name (blue). So "**value1**", "**value2**" and "**value3**"

With "OK" the Value Address is accepted and the dialog is closed.

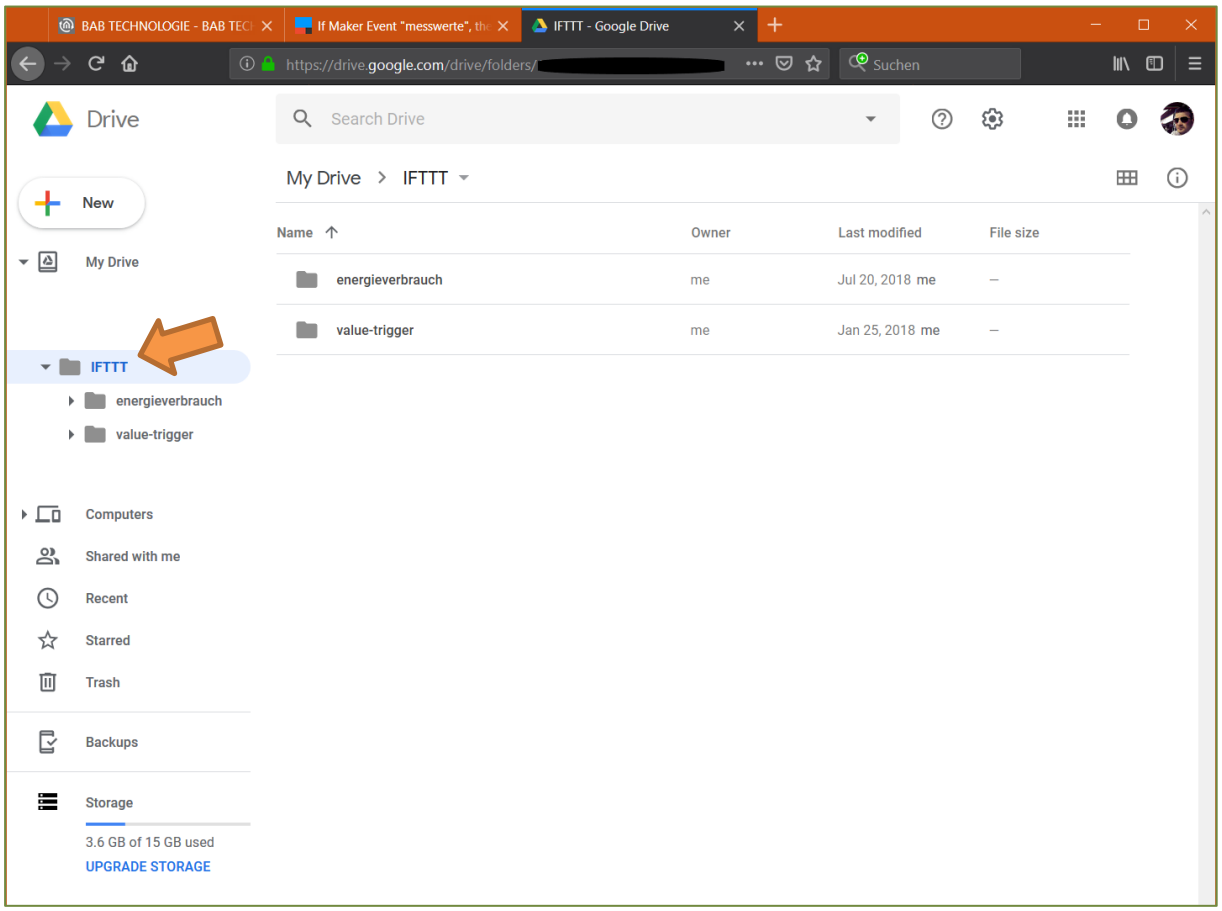
After saving via "Save instance", the instance is activated and ready for use.

TESTING THE FUNCTION

To test the function, the Google Drive is to open with a web browser. The picture shows the Google Drives.

The expected directory "messwerte" does not exist because the applet has not yet been executed!
In this figure, however, the "IFTTT" directory is already available, as this Google Drive has already been used for other IFTTT applets.

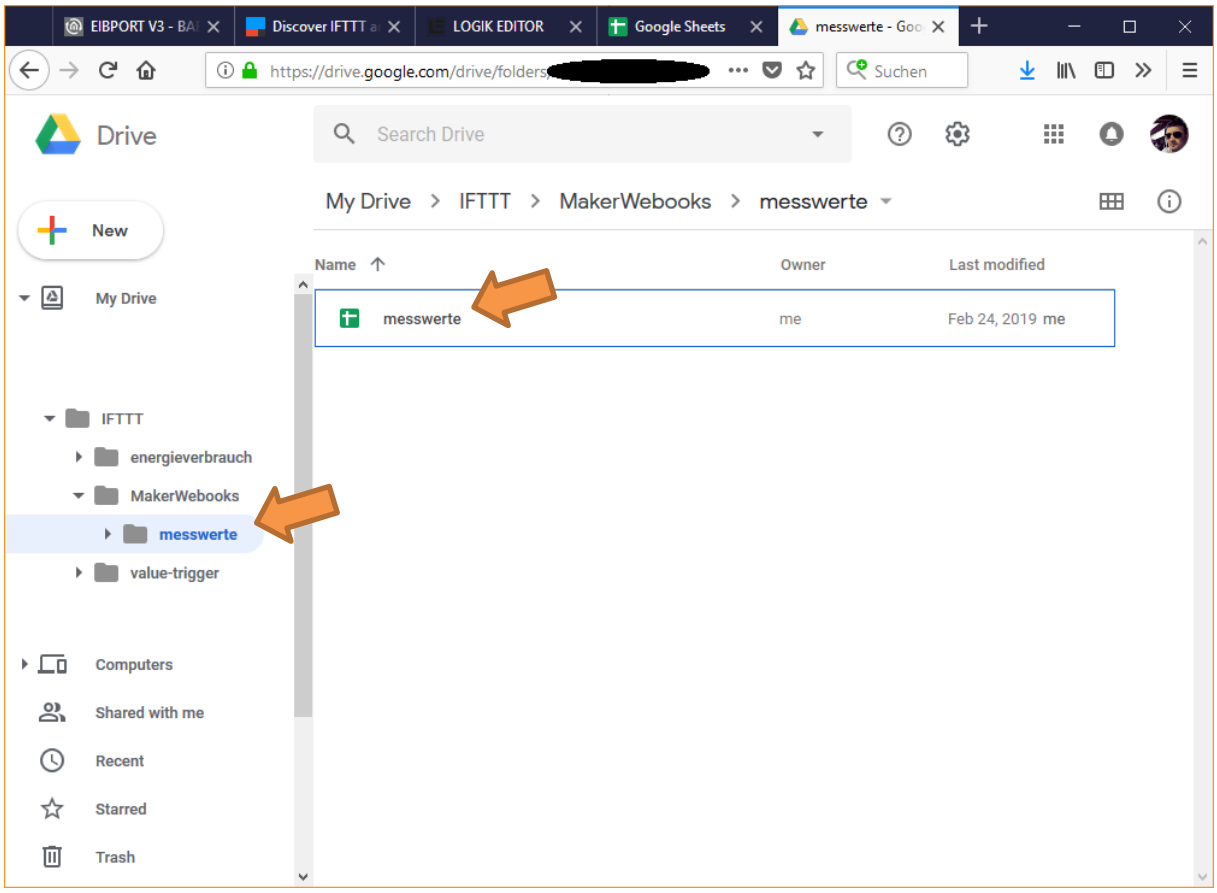
The IFTTT applet automatically creates the directory if it does not exist.



Before the IFTTT Connect instance is triggered, values must first be sent to the value addresses.

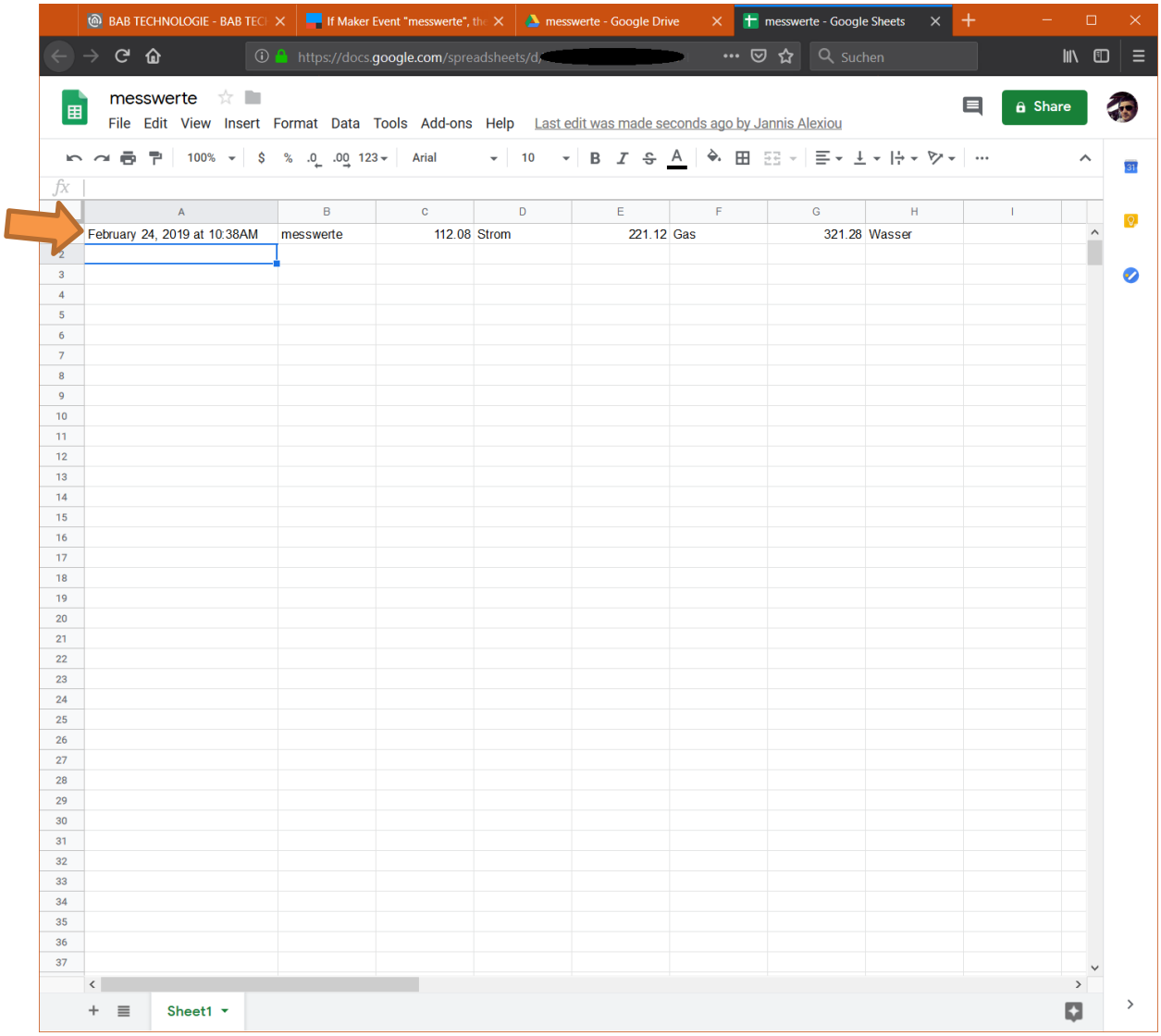
The instance is then triggered via the trigger address.

The table is now created on the Google Drive under the specified directory.



If the table is now called up, the values are entered in the table according to the specified scheme.

Another line is added to the table with every further http request from the **APP**MODULE.



The function is now tested and can be used.

7 ATTACHMENT

7.1 DATAPOINT TYPES

Function	EIS type	Data point type	Typical value	Data	Identifier
Switching	EIS 1	DPT 1.yyy	[0] = Off | FALSE; [1] = On | TRUE	1 Bit	1-bit
Relative Dimming	EIS 2	DPT 3.yyy	**„Dimming steps“**: [[0],[2…7]] Darker [2, 4, 8, 16, 32, 64] -Steps and [[1],[2…7]] Brighter [2, 4, 8, 16, 32, 64]-Steps **„Start/Stop Diming“**: [0,8] Stop; [1] Darker und [9] Brighter	4 Bit	4-bit
Time	EIS 3	DPT 10.yyy	hh:mm:ss	3 Byte	Time
Date	EIS 4	DPT 11.yyy	dd:mm:yyyy	3 Byte	Date
Floating point number (short)	EIS 5	DPT 9.yyy	-671 088,64 … 670 433,28	2 Byte	2-byte float value
Percent, Position, Brightness, …	EIS 6	DPT 5.yyy	0 … 100%	1 Byte	8-bit unsigned value
Blinds Drive/adjust	EIS 7	DPT 1.yyy	[0] = up; [1] = down When driving [0,1] = stop	1 Bit	1-bit
Priority	EIS 8	DPT 2.yyy	[0], [1] Switch on, off; [3] = Forced off; [4] = Forced on	2 Bit	1-bit controlled
IEEE Floating point number (long)	EIS 9	DPT 14.yyy	4-Octet float value; IEEE 754	32 Bit	4-byte float value
Counter 16 Bit Unsigned	EIS 10u	DPT 7.yyy	0 … 65.535	16 Bit	2-byte unsigned value
Counter 16 Bit Signed	EIS 10	DPT 8.yyy	-32.768 … 32.767	16 Bit	2-byte signed value
Counter 32 Bit Unsigned	EIS 11u	DPT 12.yyy	0 … 4.294.967.295	32 Bit	4-byte unsigned value
Counter 32 Bit Signed	EIS 11	DPT 13.yyy	-2.147.483.648 … 2.147.483.647	32 Bit	4-byte signed value
Access control	EIS 12	DPT 15.yyy	Access data	4 Byte	Entrance access
ASCII Character	EIS 13	DPT 4.yyy	Char	1 Byte	Character
Counter 8 Bit Unsigned	EIS 14u	DPT 5.yyy	0 … 255	8 Bit	8-bit unsigned value
Counter 8 Bit Signed	EIS 14	DPT 6.yyy	-128 … 127	8 Bit	8-bit signed value
String	EIS 15	DPT 16.yyy	14 Characters	14 Byte	Character string

EIB/KNX devices exchange fixed prescribed data formats with each other. These are defined in types.
The old designations of the types are EIS (EIB Interworking Standard). The new designations are DPT (Data Point Type)