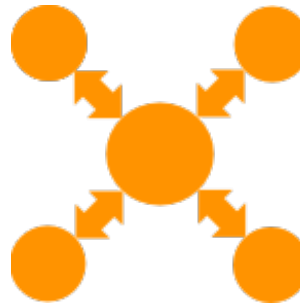
Sitelok
Webhooks Plugin



V1.1

Webhooks **Plugin**

Copyright 2023 Vibralogix. All rights reserved.

This document is provided by Vibralogix for informational purposes only to licensed users of the Sitelok product and is provided on an ‘as is’ basis without any warranties expressed or implied.

Information in this document is subject to change without notice and does not represent a commitment on the part of Vibralogix. The software described in this document is provided under a license agreement. The software may be used only in accordance with the terms of that license agreement. It is against the law to copy or use the software except as specifically allowed in the license.

It is the users responsibility to ensure the suitability of the product before using it. In no circumstances will Vibralogix be responsible for any loss or damage of data or programs as a result of using the product. Your use of the product implies acceptance of these terms.

Contents

Chapter 1 Introduction 4
What is the Webhooks plugin? 4
Chapter 2 Installation 5
Installing for the first time or upgrading 5
Disabling the Plugin 5
Uninstalling the plugin 5
Chapter 3 Setting up the plugin 6
Webhooks called when events happen in Sitelok 6
Webhook key (optional) 7
User added (sluseradded) 7
User updated (sluserupdated) 7
User deleted (sluserdeleted) 8
User login (sluserlogin) 8
User logout (sluserlogout) 8
User payment (sluserpayment) 8
Incoming actions 9
Add user (sladduser) 10
Update user (slupdateuser) 10
Add or update user (sladdorupdateuser) 10
Delete user (sldeleteuser) 10
Get user (slgetuser) 10
Chapter 4 Example using Zapier 11
Create a Google Sheet 11
Create the Zap and set the trigger 11
Get the Webhook URL 13
Choose the action 13
Connect Google Drive account and choose the spreadsheet 14
Set the fields to use 15
Testing the Zap 15
Chapter 5 Example using make.com 16
Chapter 6 Support 22

Chapter 1 Introduction

What is the Webhooks plugin?

The webhooks plugin allows Sitelok to connect with third party services such as Zapier, Integrately and Make opening up many automation possibilities such as adding new Sitelok users to a mailing service or adding a new row for each new user to a Google sheet.

Events in Sitelok such as a user being added or modified can trigger a web hook call to the service where further actions can be performed.

Also actions on that service can call back to the plugin to perform actions such as adding or modifying a user in Sitelok.

The webhooks plugin has been designed to be as flexible as possible and uses standard JSON payloads for the data tased to or from the services.

Chapter 2 Installation

Installing for the first time or upgrading

1) Extract the contents of the zip file to your PC.
2) Upload the plugin_webhooks folder to your existing Sitelok slpw folder using FTP. There are no special permissions required on most servers.
3) Login to the Sitelok control panel.
4) Open the following URL in the browser

   https://www.yoursite.com/slpw/plugin_webhooks/install.php

   which will start the installation process. If all is well you will be taken to the plugin preferences page where you will see the plugin listed.

If you have any problems with installation please let us know so that we can help you.

Disabling the Plugin

To disable the plugin select **Plugin Preferences** in the **Plugin** menu option of Sitelok. Uncheck the enable box for the plugin and click the **Save** button. You can enable it again in the same way.

Uninstalling the plugin

To permanently remove the plugin and it's settings follow these steps.

1) Disable the plugin as above.
2) Click the delete icon next the plugin in the disabled plugins section.
3) Confirm the action in the alert box.

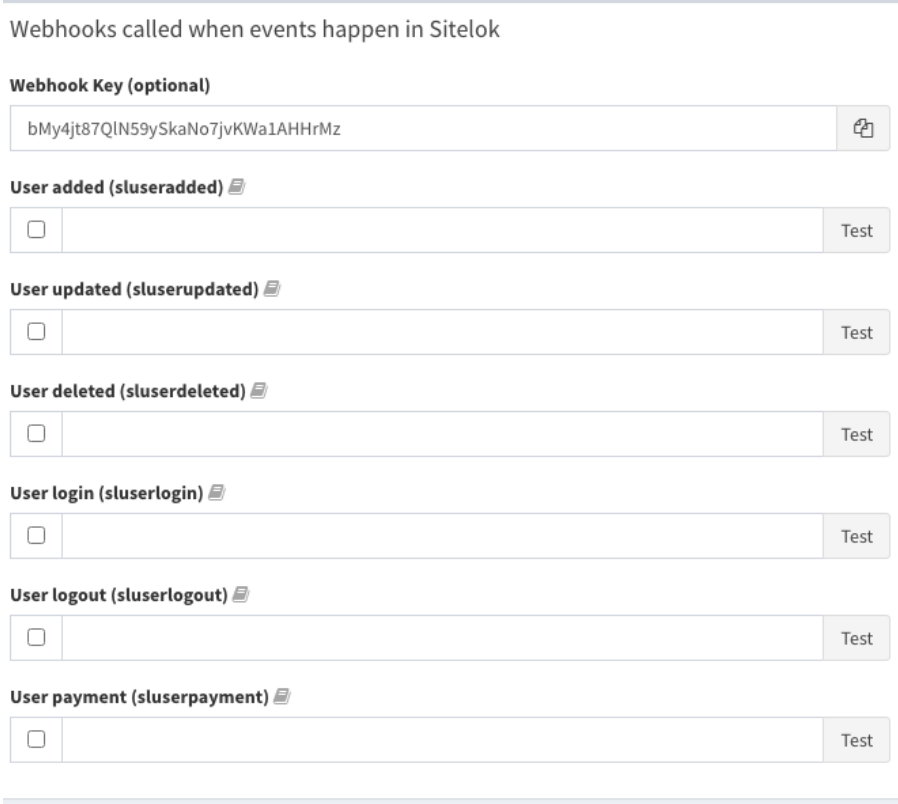
If the plugin is uninstalled successfully you will be returned to the plugin preferences page.

Chapter 3 Setting up the plugin

Click the Webhook icon in the plugins menu to access the plugin settings.

You will see that there are two main sections. The first sections controls the outgoing webhook calls that occur when events happen in Sitelok. The second section controls the incoming action which are webhook calls that cause actions to happen in Sitelok.

Webhooks called when events happen in Sitelok

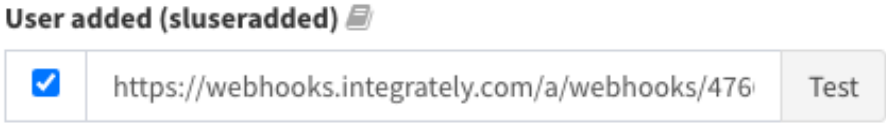


In this section you can set URL's to call for each event type and also enable or disable it. There is an options web hook key that get sent with the call which you can use to verify the call cam from the plugin (if required).

Webhook key (optional)

Services such as Zapier and Integrately use a unique secret URL which is unknown to others but you can verify the optional webhook key for extra security if required.

The next fields allow you to enable and set a URL for each event type. For example here we have set an Integrately web hook URL that will be called when users get added to Sitelok.



You can click the click the icon to access the documentation for the call which includes example JSON data and explanations of each field. When setting up web hooks in services like Zapier and Integrately you can click the test button to send example data. This then allows you to select the data fields that you need. Data is sent as JSON.

User added (sluseradded)

If you set a URL in this field (and enable it via the checkbox) it will be called when a user is added to Sitelok. All of the users fields are sent (except password). Please note this event is not called for bulk actions such as importing users.

User updated (sluserupdated)

If you set a URL in this field (and enable it via the checkbox) it will be called when a user is updated in Sitelok. All of the users fields are sent (except password). The previous field contents and which fields have changed are also sent in the JSON data. Check the online documentation for example data and field descriptions.

By default this web hook is called whenever any changes to the user are made. However you can instead have the call made only when certain fields are updated (for example their email changes). To do this select Call

web hook only when these fields change in the drop down and check the required fields.

Please note this event is not called for bulk actions such as importing users or canes mad via the maintenance plugin.

User deleted (sluserdeleted)

If you set a URL in this field (and enable it via the checkbox) it will be called when a user is deleted in Sitelok. All of the users fields are sent (except password).

User login (sluserlogin)

If you set a URL in this field (and enable it via the checkbox) it will be called when a user logs in to Sitelok. All of the users fields are sent (except password).

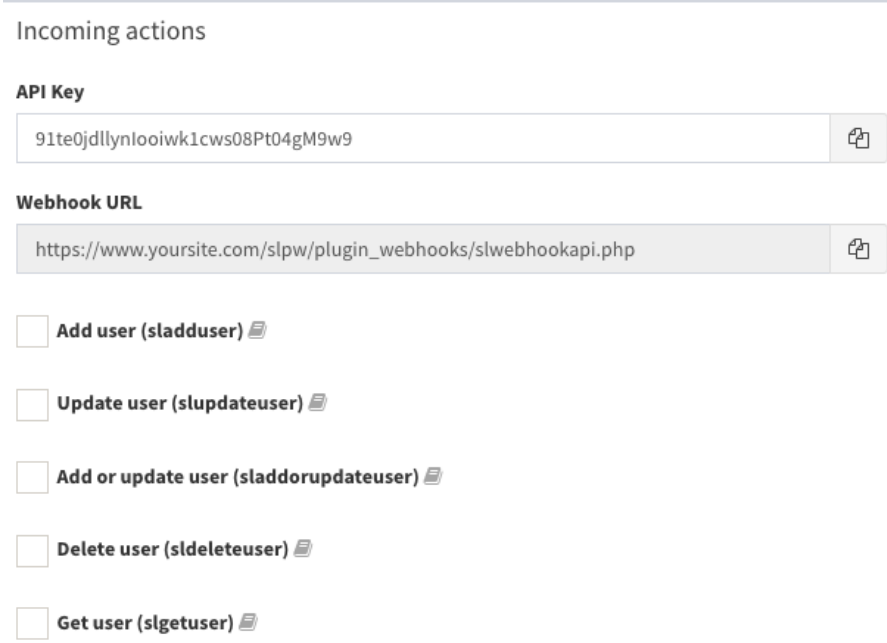
User logout (sluserlogout)

If you set a URL in this field (and enable it via the checkbox) it will be called when a user logs out of Sitelok. All of the users fields are sent (except password).

User payment (sluserpayment)

If you set a URL in this field (and enable it via the checkbox) it will be called when a user makes a payment via a payment plugin. All of the users fields are sent (except password) along with the payment details.

Incoming actions



In this section you can see the API key and Webhook URL that shall be used for incoming web hook calls. You can also enable or disable each of the incoming web hook functions. We recommend, for security, only to enable the functions you need.

You can click the click the icon to access online documentation for each function. Included are example JSON data that should be sent to the Webhook URL and descriptions of each field. Note that you can have the webhook send an email sent to the user and/or the site admin by setting the template file name in the properties

```
"useremailtemplate": "newuser.htm",
"adminemailtemplate": "newuseradmin.htm"
```

Note that although similar to a REST API you should still use the POST method for all of the functions (not PUT, DELETE etc). You can send the api key either in the Authorization header or by using the apikey field in the JSON.

Add user (sladduser)

This webhook API function can be called to add a new user to Sitelok. For security you cannot add users in the ADMIN or SUBADMIN usergroups.

Update user (slupdateuser)

This webhook API function can be called to update an existing Sitelok user. All fields can be modified apart from the password. For security you cannot add the ADMIN or SUBADMIN userrgroups to the user though.

Add or update user (sladdorupdateuser)

This webhook API function can be called to modify an existing users data. If the user doesn't exist then a new user is created. This can be useful where you dont know if a user already exisits or not.

Delete user (sldeleteuser)

This webhook API function can be called to delete a user. For obvious reasons you need to be careful if you enable this function.

Get user (slgetuser)

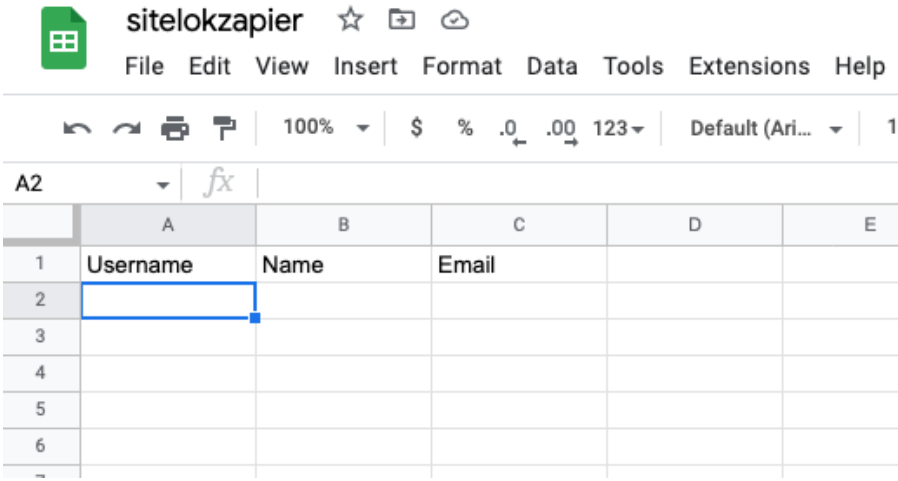
This webhook API function can be called to retrieve a specific users data. All fields are returned except for the password.

Chapter 4 Example using Zapier

This example uses Zapier to add a new row to a Google Sheet when a new user is added to Sitelok (Sitelok calling Zapier). Integrately and other services can be setup in a similar way.

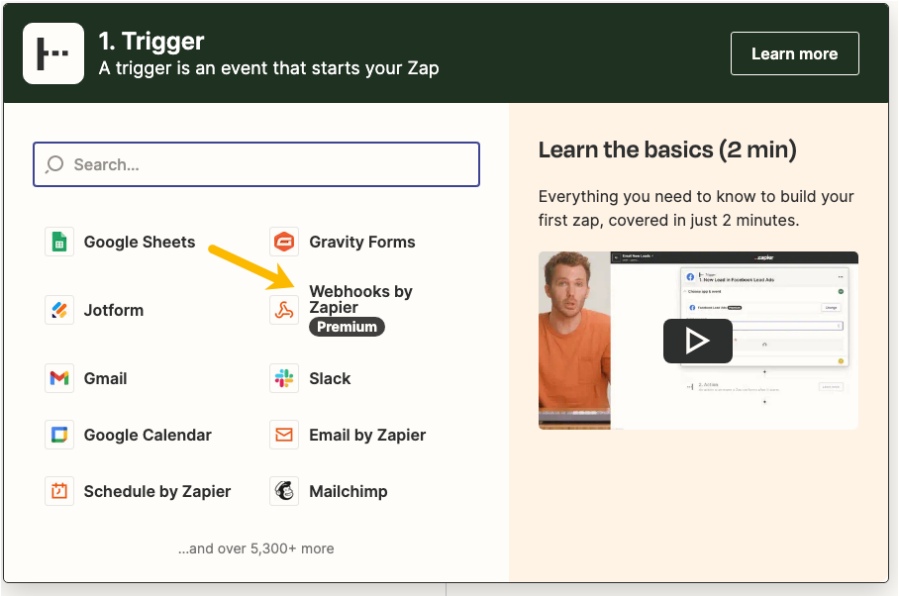
Create a Google Sheet

First of all we need to create a Google Sheet to collect the username, name and email of each user. Enter the column headers in the first row.

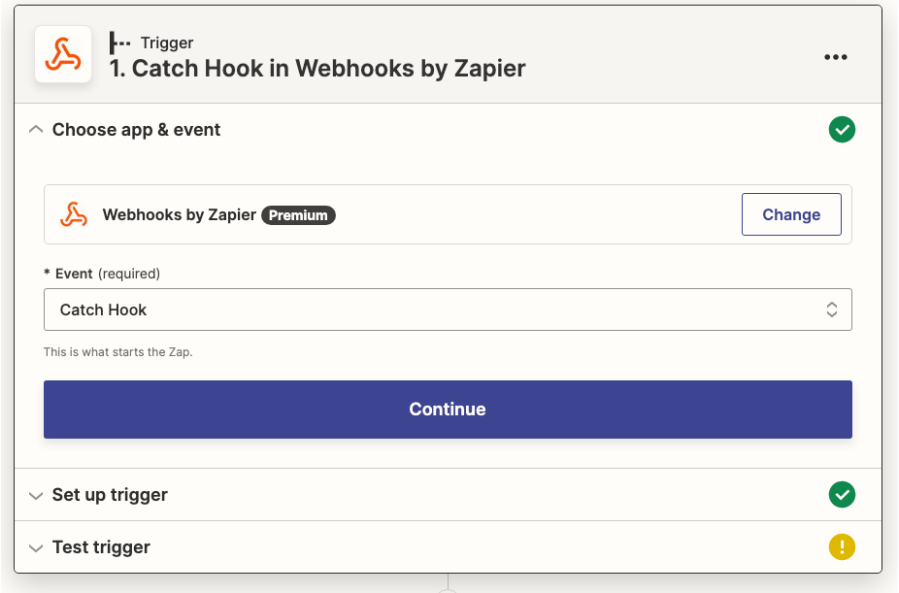


Create the Zap and set the trigger

In your Zapier account create a new Zap. We will set the trigger to be a Webhook.

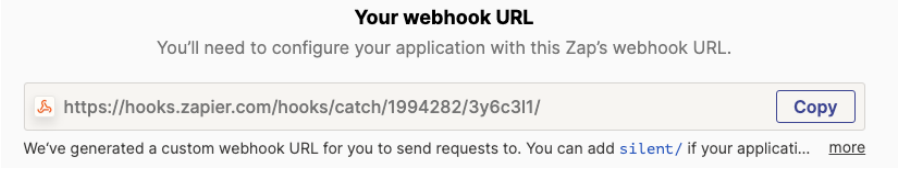


Click Webhooks by Zapier and select Catch Hook as the Event.



Get the Webhook URL

Click Test Trigger and copy the URL.

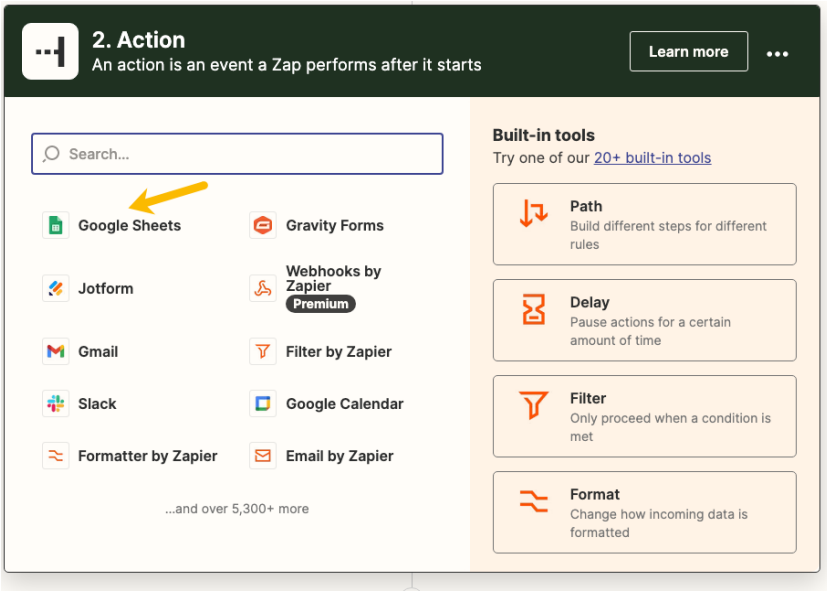


Open up the Sitelok Webhook plugin and paste in the Zapier web hook URL in the User added event field and save the settings.

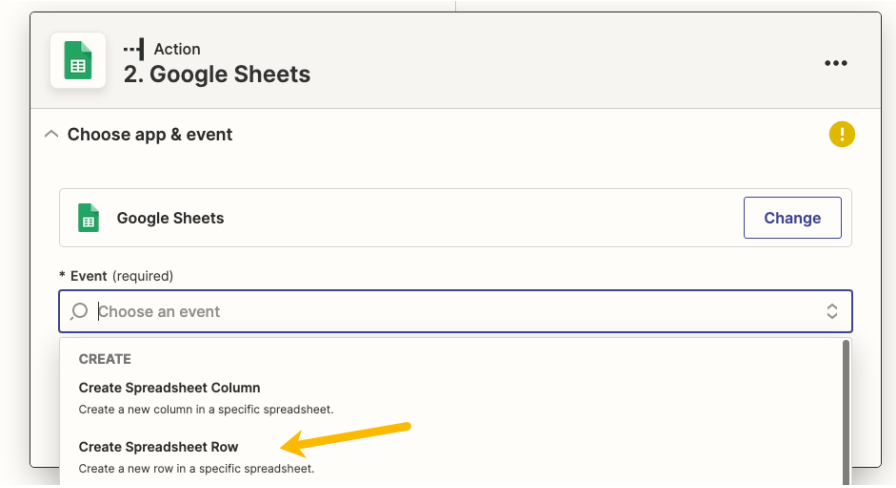
Click the Test button to send test data to the URL.

Choose the action

In Zapier click Continue and select Google Sheets as the action

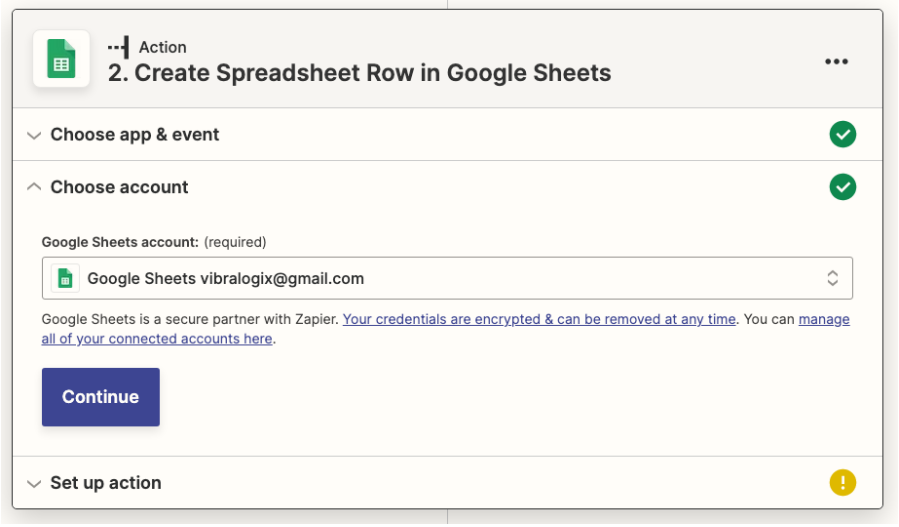


We now set the event to be Create Spreadsheet Row



Connect Google Drive account and choose the spreadsheet

In the Choose account section you can connect to your Google Driveraccount (or select and already connected account).

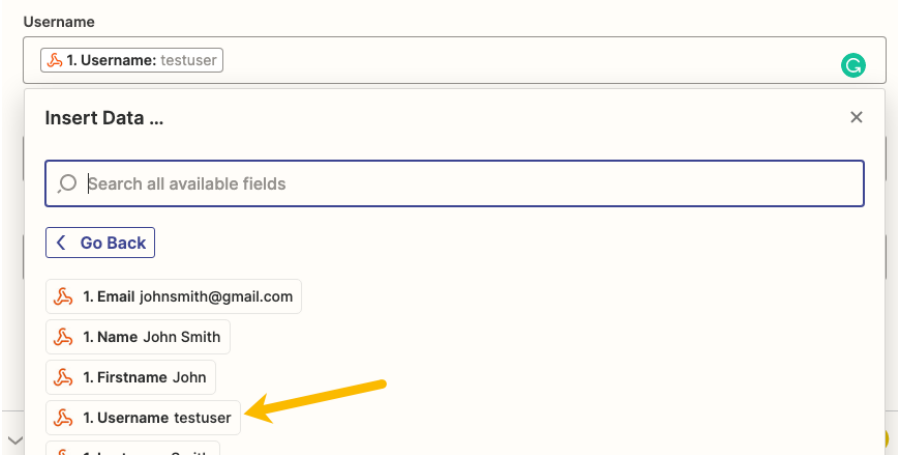


Select the Spreadsheet and worksheet to use.

Set the fields to use

Once you choose your spreadsheet you should see the columns headers appear. For each one you can set the data to insert.

Click in the Username field. Zapier should have received the example text call (when you clicked Test in the plugin) so should show the field list and example data for you to chose from.Select username. If not just enter username (which is the field we want to store in this column).



Repeat the above for the Name column (name field) and Email column (email field).

Testing the Zap

In the Test action section click the Test button

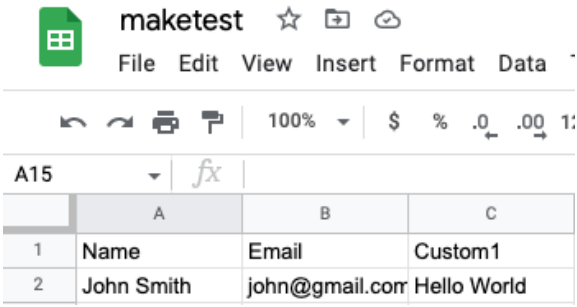
If everything was setup correctly you should see the example data added to your spreadsheet so its ready to publish.

	A	B	C	D
1	Username	Name	Email	
2	testuser	John Smith	johnsmith@gmail.com	

Chapter 5 Example using make.com

This example shows how to use make.com to call Sitelok (incoming action) to add a user when a new row is added to a Google Sheet.

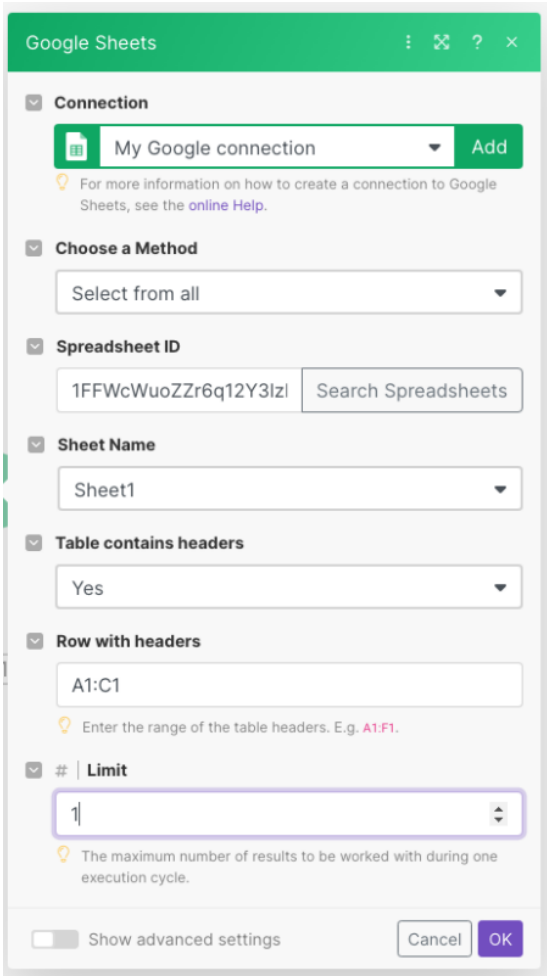
I setup a Google sheet like this



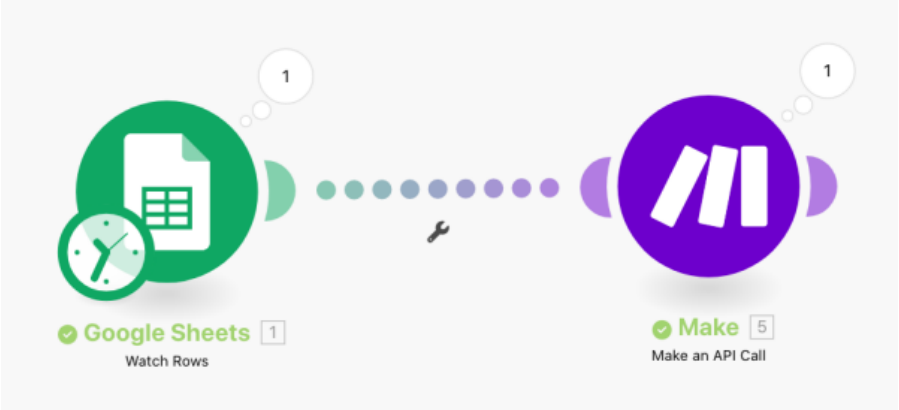
In Make create a new scenario and add a Google sheet watch rows module setup as in the next image.

During the process you will need to connect make to your Google account and select the sheet to use.

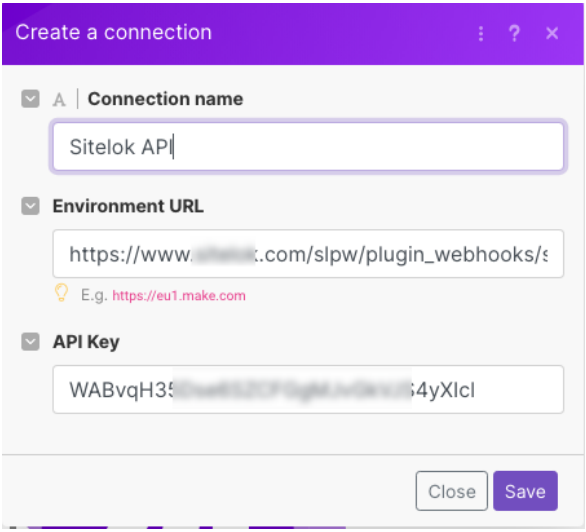
You will also need to specify the column header cells (A1:C1 in this example).


Google Sheets
Connection
My Google connection
Add
For more information on how to create a connection to Google Sheets, see the online Help.
Choose a Method
Select from all
Spreadsheet ID
1FFWcWuoZZr6q12Y3lzl
Search Spreadsheets
Sheet Name
Sheet1
Table contains headers
Yes
Row with headers
A1:C1
Enter the range of the table headers. E.g. A1:F1.
| Limit
1
The maximum number of results to be worked with during one execution cycle.
Show advanced settings
Cancel
OK

Now we need to add a new module to make the call back to Sitelok. Choose the Make module and select the Make an API call option.



In the Make API module you will need to make a new API connection like this.



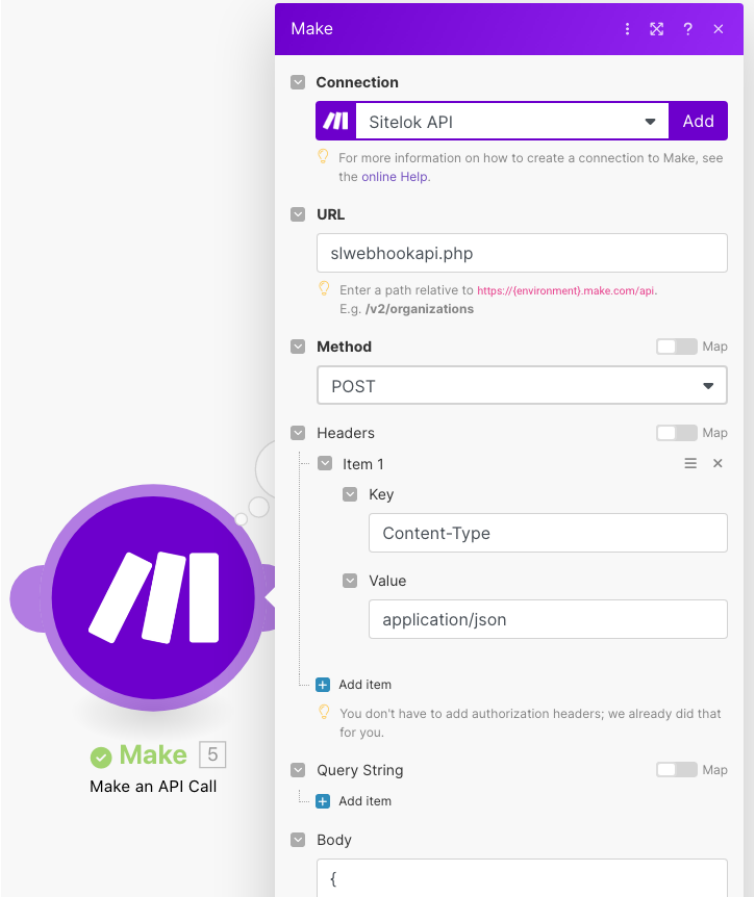
For the environment URL use the full Webhook URL from the plugin page (for example)

https://www.yoursite.com/slpw/plugin_webhooks/slwebhookapi.php

For the API key use the API key from the plugin page.

When you save the connection it will verify the connection. Note that this connection can be used for other scenarios as well.

Set up the rest of the Make API Call like this. For the URL just use slwebhookapi.php (without the rest of the URL)

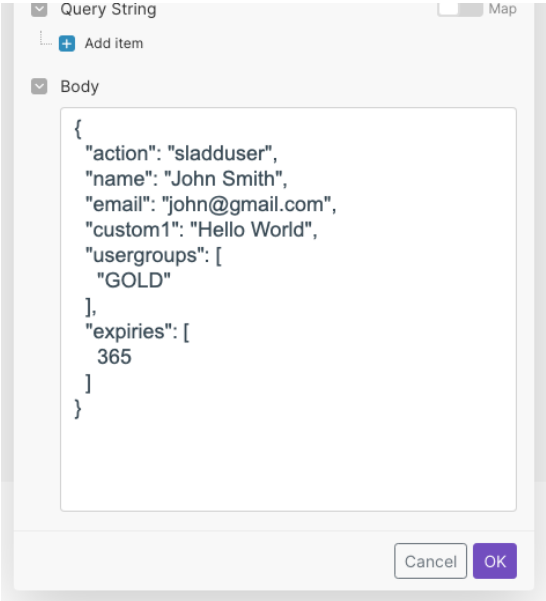


Now we need to make the JSON data to send to the API. Here is an example that will add a new user (see the plugin docs links).

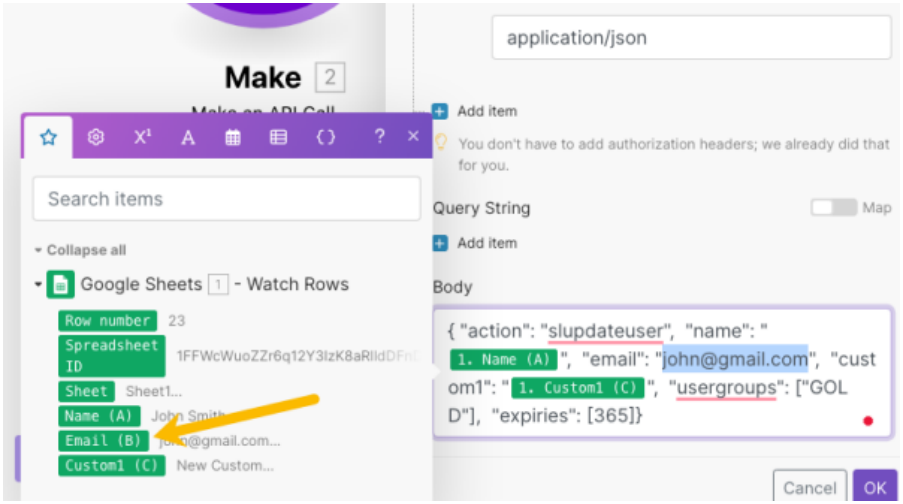
```
{
  "action": "sladduser",
  "name": "John Smith",
  "email": "john@gmail.com",
  "custom1": "Hello World",
  "usergroups": [
    "GOLD"
  ],
  "expiries": [
    365
  ]
}
```

The apikey field is not needed here as Make will send it as part of the connection you setup earlier.

Paste the JSON in the Body section.

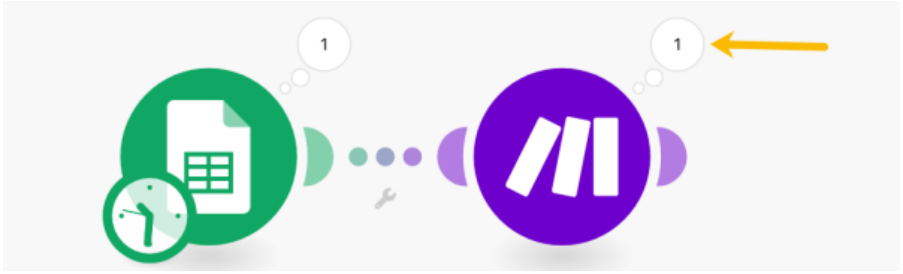


Now of course we don't want to always add John Smith so we need to change the fields we want to the data from the canned row. To do this select the data you want to replace and then choose the column field from the list. For example here we are setting the email from the Email column of the sheet.



Once you have everything setup you can add a row to your Google Sheet and then click the Run Once button in Make to test it.

If you don't see the user added in Sitelok you can click the circle above the Make module to see more information.



Chapter 6 Support

Hopefully if you have followed this manual carefully everything will be working fine. However sometimes things don't go quite so smoothly so if you have any questions or problems then please check the FAQ on the support page or email us.

Support area: https://www.vibralogix.com/support/

Email: support@vibralogix.com