



Webhooks Integration Guide

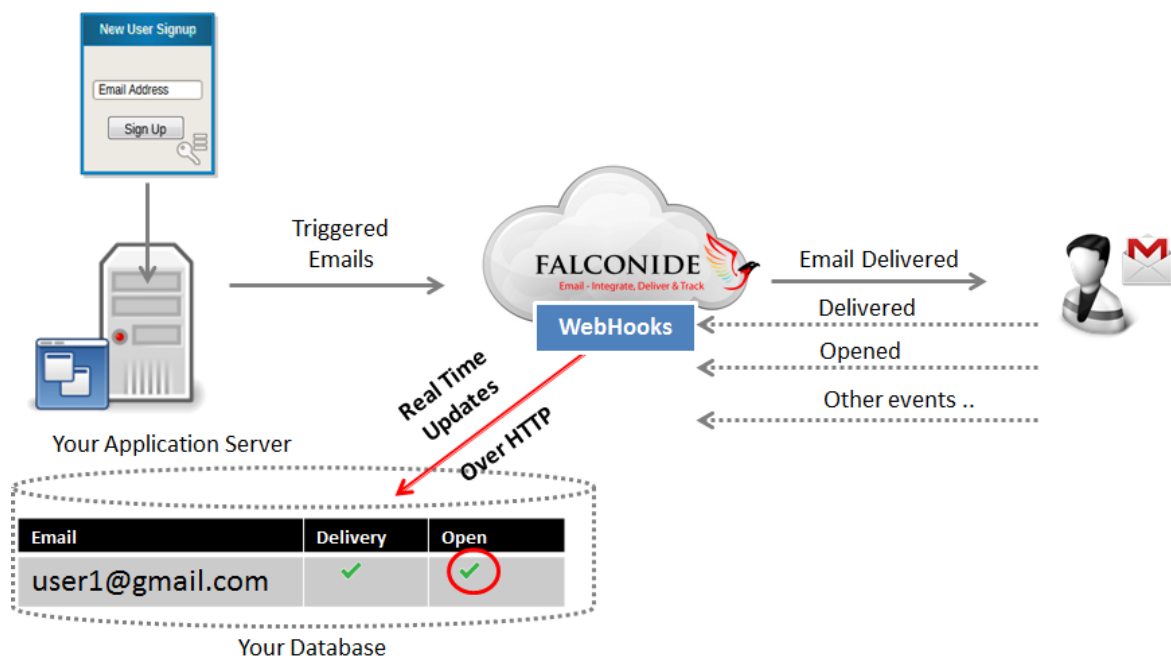
Version 2.1

Contents

1. Overview	3
2. How does Data transmission happen?	4
3. How to Configure Webhooks	5
4. Events Data	12
5. How to collect the event data?	14
6. Track your Webhooks	14
7. Support.....	14

1. Overview

Webhook is a feature in Falconide that enables real-time data syncing between Falconide and your system. A Webhook is an HTTP callback: an HTTP POST that occurs when an event like email open or click occurs and sends back information. Instead of having to poll our API constantly to check for updates, Falconide sends the data to your system at the instant an event occurs.



2. How does Data transmission happen?

When any of the events out of the ones you have specified occurs, a call back URL sends an HTTP POST request to the URL you have specified and delivers the events to your end point. If you haven't specified any URL, or the URL takes too long to respond (more than 60 seconds), the Webhook call is considered failed with no retries are allowed. Such failed calls are logged in to the database, and you can download the log from the panel. A notification is sent to the email address you specify. Data is shared over HTTP POST to your preconfigured Webhooks URL.

You may use a HTTP or a HTTPS URL for Webhooks. In most cases HTTP is sufficient, but HTTPS can be useful if your data is sensitive or if you wish to protect against replay attacks for example.

Falconide Webhooks support the following events:

- **sent:** when the email is delivered to the receiving server.
- **dropped:** when the email address is dropped. Users who have unsubscribed in past or are blacklisted because of hard bounce/Spam complaints are treated as dropped. Once dropped, no email communication is allowed on these ids.
- **invalid:** when any event is not processed due to invalid or incorrect email ID. All API requests with syntactically incorrect email ids will be treated as Invalid and no further processing will be done on such ids. You can capture all such invalid events in real time and try correcting them offline.
- **bounced:** when the email bounces, that is, the receiving server could not or does not accept message due to reasons like server not being reachable, or email id doesn't exists etc. Hard bounces from an email address lead to its blacklisting in Falconide system.
- **opened:** when a recipient opens the email
- **clicked:** when a recipient clicks a link within the email message
- **unsubscribed:** when a recipient clicks the unsubscribe link/button.

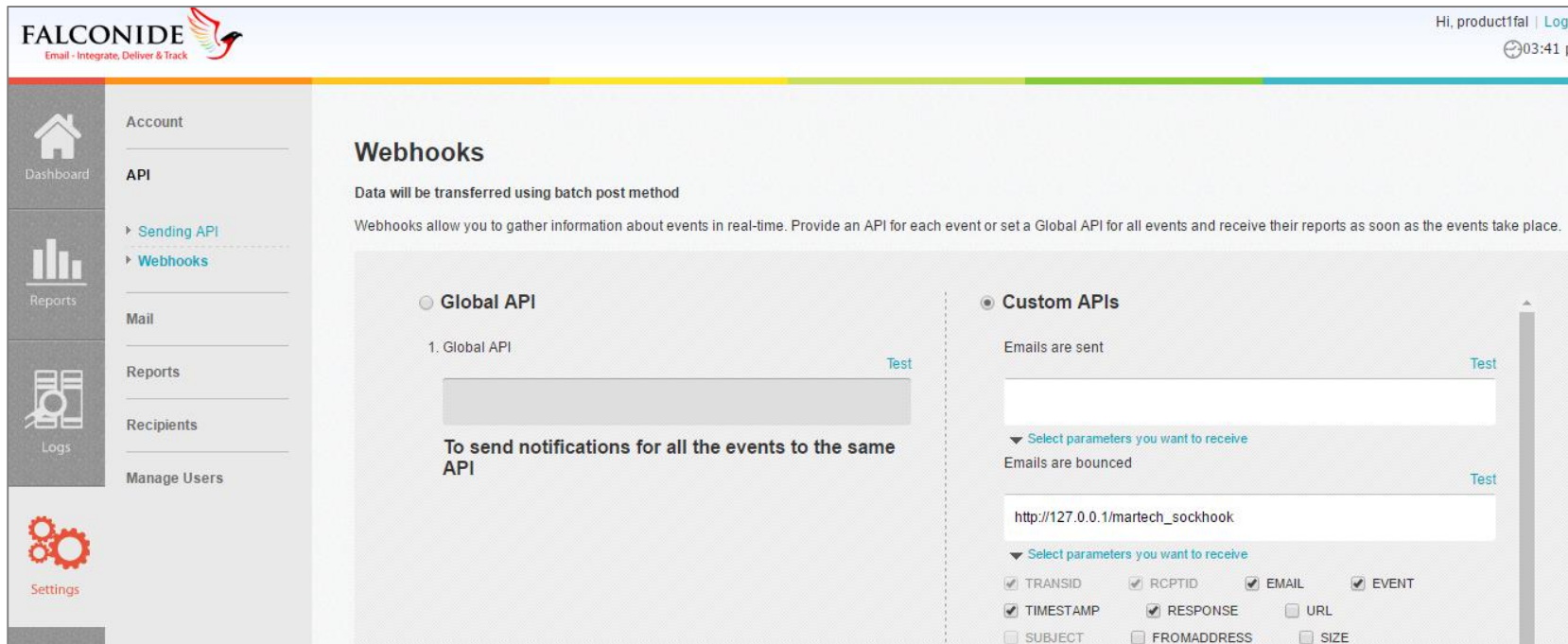
- **abuse**: when a recipient marks the email as spam.

3. How to Configure Webhooks

You can configure webhooks either from your Falconide panel, or by using APIs.

Configure webhooks from your panel:

- Log in to your Falconide account.
- Go to **Settings** -> **API**.
- Click **Webhooks** under the **API** menu.



The screenshot shows the Falconide web interface for configuring webhooks. The top navigation bar includes the Falconide logo and the tagline "Email - Integrate, Deliver & Track". The user is logged in as "product1fal" and the time is 03:41. The left sidebar contains a menu with options: Dashboard, Reports, Logs, and Settings. The main content area is titled "Webhooks" and includes a note: "Data will be transferred using batch post method". Below this, a description states: "Webhooks allow you to gather information about events in real-time. Provide an API for each event or set a Global API for all events and receive their reports as soon as the events take place." The interface is divided into two sections: "Global API" and "Custom APIs". The "Global API" section has a radio button selected and a text input field for a single API endpoint, with a "Test" button. The "Custom APIs" section has a radio button selected and is divided into two sub-sections: "Emails are sent" and "Emails are bounced". Each sub-section has a text input field for the API endpoint and a "Test" button. Below the "Emails are sent" section, there is a dropdown menu for "Select parameters you want to receive" with checkboxes for TRANSID, RCPTID, EMAIL, and EVENT. Below the "Emails are bounced" section, there is a dropdown menu for "Select parameters you want to receive" with checkboxes for TIMESTAMP, RESPONSE, URL, SUBJECT, FROMADDRESS, and SIZE.

- You can either choose to have one global API that sends notifications for all the events, or you can define custom APIs for different events.

- e. Whichever option you choose, enter a valid Call back URL where Falconide can POST the event data.
- f. If you select custom APIs, then select the parameters that you want to receive for each API.
- g. If you want to test the Webhook call, click **Test**.

Sample of JSON API Push:

```
[
{
  "TRANSID": "14507226013384557",
  "X-APIHEADER": "1234",
  "RESPONSE": "smtp;250 2.0.0 OK 1450791482
n83si34501442qhn.97 -
gsmt",
  "EVENT": "sent",
  "RCPTID": "19556",
  "EMAIL": "xxxxxxxxxx@gmail.com",
  "TIMESTAMP": "1450791482"
},
{
  "TRANSID": "14507226013383534",
  "X-APIHEADER": "1234",
  "RESPONSE": "smtp;250 2.0.0 OK 1450791482
v72si34546256qka.96 -
gsmt",
  "EVENT": "sent",
  "RCPTID": "19556",
  "EMAIL": "xxxxxxxxxx@gmail.com",
  "TIMESTAMP": "1450791482"
},
{
  "TRANSID": "14507226013385581",
  "X-APIHEADER": "1234",
  "RESPONSE": "smtp;250 2.0.0 OK 1450791482
m11si34606295qgd.21 -
gsmt",
  "EVENT": "sent",
  "RCPTID": "19556",
  "EMAIL": "xxxxxxxxxx@gmail.com",
  "TIMESTAMP": "1450791482"
}
]
```

Important:

1. If your system is behind a firewall that blocks access to all traffic except for certain domains, then you must grant access to **Falconide.com** so that the Webhooks can function. You need to allow **202.162.242.179/255** range of IPs on your network.
2. Your Webhooks URLs should be set up to accept the POST requests coming from the Falconide server. When you provide the URL where you want Falconide to POST the data for events, a quick check is done through HEAD request (not POST) to ensure that the URL exists.

If the URL doesn't exist or returns something other than a 200 HTTP response to the HEAD request, Falconide will fallback and attempt a POST request. The POST is the same type of POST as a Falconide Webhook, except that the parameter is an empty array.

Configure webhooks through APIs:

To add a webhook, use the following API:

Input parameters:

- a. **API Key**: API credentials provided by Falconide.
- b. **type**: It defines the module.
- c. **URL**: The call back URL where Falconide can POST the event data.
- d. **action**: You need to define the event here, for which you want to receive the information when that respective event happens.
- e. **params**: Parameters which you want to receive in all API calls via Webhooks.

Input JSON:

<http://api.falconide.com/falconapi/manage.falconide?>

[data=](#)

```
{ "api_key" : "<api key>",  
  "type" : "addwebhook",  
  "url": "<valid http url with allowed post request>",  
  "action" : "<sent/opened/clicked/bounced/dropped/invalid/unsub/abuse/dropped>", //Provide the action that  
  you want the url to perform  
  "params":{"timestamp":"<value>", "event":"<value>", "response":"<value>", "fromaddress":"<value>",  
  "size":"<value>","tags":"<value>"} //To fetch the data of the parameter, provide the value as '1', else  
  '0'  
}
```

Sample Input JSON:

<http://api.falconide.com/falconapi/manage.falconide?>

```
data=  
{  
  "api_key" : "43dd4743ad54b20eb446b060289df555",  
  "type" : "addwebhook",  
  "url": "http://emml6.netcore.co.in/ip.php <http://emml6.netcore.co.in/ip.php>,  
  "action" : "opened",  
  "params":{"timestamp":"0", "event":"0", "response":"1", "fromaddress":"0", "size":"0", "tags":"0"}  
}
```

Important:

- If you do not pass any value to the 'action' parameter, then the URL is treated as the Global API URL.
- If you do not specify which parameters you require, then by default, values for all the parameters are passed in API.
- XAPI-HEADER is passed as a parameter by default.

Output JSON

Success:

```
{  
  "status": "success",  
  "message": "webhook api integrated successfully."  
}
```

Failure:

```
{  
  "status": "error",  
  "message": "Missing webhook url"  
}
```

To remove a webhook , use the following API:

Input JSON:

<http://api.falconide.com/falconapi/manage.falconide?>

```
data=
{
  'api_key' : '<api key>',
  'type' : 'removewebhook',
  'action': '<sent/opened/clicked/bounced/dropped/invalid/unsub/abuse/dropped>' } // Provide the action that
you want the url to perform
```

Sample Input JSON:

<http://api.falconide.com/falconapi/manage.falconiden?>

```
data=
{
  'api_key' : '43dd4743ad54b20eb446b060289df555',
  'type' : 'removewebhook',
  'action': 'sent' }
```

Important:

If you do not specify any action, all the webhook URLs are removed.

Output JSON:**Success:**

```
{  
  "status": "success",  
  "message": " webhook api removed successfully."  
}
```

Failure:

```
{  
  "status": "error",  
  "message": " error removing webhook settings"  
}
```

4. Events Data

Here is a sample data for each event showing the actual fields that are returned in the HTTP POST:

Parameters	Description	Supported Events
TRANSID	The unique transaction ID that Falconide assigns to each sent email	All
EMAIL	The email address of the recipient	All
EVENT	Type of Event: sent/ dropped/ invalid/ bounced/ opened/ clicked/ unsubscribed/ abuse	All
RESPONSE	Response received from the end server: Delivery Logs, Bounce reason, reason for drop (blacklisted user / already unsubscribed)	sent, dropped, invalid, bounce
X-APIHEADER	Information passed by you in the X-APIHEADER, during the time of email sent	All
TIMESTAMP	Unix Timestamp of the occurrence of the event	All
USERAGENT	Detailed information about the browser from where a specific event is initiated	opened, clicked, unsubscribed, abuse
IPADDRESS	IP address of the device from which the recipient responds	opened, clicked, unsubscribed, abuse
SUBJECT	Subject of the email	All
FROMADDRESS	From address which is used in email at the time of email sent	All

Here are samples of values that are returned for the events:

Events	Sample values												
	TRANSID	EMAIL	RESPONSE	X-APIHEADER	TIMESTAMP	USERAGENT	IPADDRESS	SUBJECT	FROM ADDRESS	BOUNCE_TYPE	BOUNCE_REASON	BOUNCE_REASONID	URL
sent	198987967526	test@example.com	175.158.64.39-> 250 2.0.0 OK k2si444142...	ACC12312	1358402419	NA	NA	Thank you email	info@example.com				
dropped	198987967526	test@example.com	Blacklisted user / Already unsubscribed	ACC12312	1358402419	NA	NA	Thank you email	info@example.com				
invalid	198987967526	test@example.com	Invalid Email Address	ACC12312	1358402419	NA	NA	Thank you email	info@example.com				
bounced	198987967556	test@example.com		ACC12311	1358402419	NA	NA	Thank you email	info@example.com	HARDBOUNCE /SOFTBOUNCE	550 5.1.1 The email account does not exist	342	
opened	1989867556	test@example.com	NA	ACC12311	1358402419	Mozilla/5.0 (Macintosh; Intel AppleWebKit/537.36	209.167.18.19	Thank you email	info@example.com				
clicked	1982796	test@example.com	NA	BC12311	1358402419	Mozilla/5.0 (Macintosh; AppleWebKit/537.36	209.167.18.19	Thank you email	info@example.com				http://www.wearewater.org/index.php
unsubscribed	1987556	test@example.com	NA	AC12311	1358402419	Mozilla/5.0 (Macintosh; Intel AppleWebKit/537.36	209.167.18.19	Thank you email	info@example.com				

abuse	198987967556	test@gmail.com		ACC12311	1358402419	NA	NA	Thankyou email	info@example.com				
-------	--------------	----------------	--	----------	------------	----	----	----------------	------------------	--	--	--	--

5. How to collect the event data?

To collect the event data posted by Webhooks, you need to write a script in any programming language of your choice. This is language as well as platform independent. Here is a sample PHP script:

```
<?php
$fp =fopen ('/tmp/weblog.txt', 'a');
foreach($_POST as $key=>$val) {
    fwrite($fp, "$key=>$val");
}
fwrite($fp, "\n");
fclose($fp);
?>
```

Sample Function to convert Timestamp into Date:

```
echo date('m/d/Y', 1299446702);
```

6. Track your Webhooks

All the Webhook calls are logged in real-time. You can view/download the reports from the Reports section in the Falconide panel.

7. Support

If you have a question, call **+91-(22)-66628081**

You can also drop a mail to **support@falconide.com**