



Documentation

E-Marketing Manager

Webhooks

Version 1.1.4
2025-09-10

Publisher:
AGNITAS AG
Werner-Eckert-Strasse 6
D-81829 Munich

At the end of the document you will find a changelog where you can retrace all amendments compared with the previous version.

Table of Contents

1	Introduction to webhook interface	4
2	Terminology	4
3	Basics on notification mechanism	4
4	Webhook endpoints	5
5	Recipient data	6
6	Message structure	6
6.1	General.....	6
6.2	Top level structure	6
6.3	Basic event structure	6
6.4	Examples.....	8
6.4.1	mailing_opened (one event)	8
6.4.2	mailing_opened (multiple events)	8
6.4.3	link_clicked.....	9
7	Events	10
7.1	Mailing delivery	10
7.1.1	mailing_delivered.....	10
7.1.2	hard_bounce.....	11
7.1.3	mailing_delivery_complete.....	11
7.2	Recipient reactions.....	12
7.2.1	link_clicked.....	12
7.2.2	mailing_opened.....	12
7.2.3	binding_changed.....	13
7.2.4	profile_field_changed.....	14
8	Tables	15

8.1	User states	15
8.2	Media types	16
9	Changelog	16

1 Introduction to webhook interface

The webhook interface allows clients to receive notifications on events on mailings, recipients, etc.

2 Terminology

- **Endpoint:** see *webhook endpoint*
- **Event:** Something, that happens in EMM like: recipient opened a mailing or mailing has been delivered successfully
- **Event data:** Information on a single event. Event data is contained in a webhook message.
- **Notification:** See *webhook notification*
- **Webhook:** Interface that sends notifications on events to a given URL. Each webhook is responsible for one event type.
- **Webhook endpoint:** Receiving service of webhook notifications.
- **Webhook message:** A webhook message contains one or more *event data* items of same type.
- **Webhook notification:** synonym for *webhook message*.

3 Basics on notification mechanism

Each time an event occurs, data on this event is sent to a webhook endpoint.

To save resources on both sides, notifications are not sent immediately when the event occurs. Event data are collected and sent after a certain period of time. A notification contains event data of one event type.

The webhook endpoint must acknowledge received notifications correctly, otherwise event data will be resent after a certain period of time. Event data will be discarded after 10 unsuccessful notification attempts.

4 Webhook endpoints

Protocol

All webhook endpoints must be accessible by HTTPS protocol. HTTP or other protocols are not allowed.

Acknowledging webhook notifications and timeouts

When receiving a webhook notification, the endpoint must respond with HTTP 200 within 10 seconds in order to acknowledge all events in notification. All other HTTP response codes or timeouts are treated as an error indicator and will resend event data later.

Note: Redirection responses (HTTP 3xx, especially 301 "Moved permanently", 302 "Moved temporarily", 307 "Temporary redirect" and 308 "Permanent redirect") are not interpreted and considered as error indicators.

Resent event data

Event data, that is not acknowledged correctly by the webhook endpoint is resent after a period of time. The endpoint must be able to deal with resent event data. The endpoint can use the event ID contained in each event data to identify such data.

Resent event data may be combined with other event data in a webhook message.

Event timestamps

Due to several reasons (different event types, resent event data, ...), event data cannot be sent in chronological order in all cases. For such cases, each event data is augmented with a timestamp to indicate when the event occurred.

Retrieving additional information

Event data in webhook notifications provide a set of basic common information.

Additional data can be retrieved by SOAP webservises or Rest API. It is highly recommended to store data locally, that has already been retrieved and use SOAP-/Rest API only if needed.

5 Recipient data

Some notifications can include data on the involved recipient. This can be just the ID of the recipient (property "recipient_id") or an additional set of profile fields (property "recipient_data").

The set of profile fields can be configured differently for each event type.

Tracking veto

Recipients can decide not to be tracked. In such a case, the recipient ID included in the notification data shows "not_tracked" and the "recipient_data" property is omitted.

6 Message structure

6.1 General

- Timestamps are given in UTC timezone. The format is "YYYY-MM-DD'T'HH:MM:SS'Z'". (Example: 2021-03-29T07:56:32Z)

6.2 Top level structure

Message data is sent as a JSON structure. At top-level, the JSON is

```
{
  "event_count": <number of events>,
  "event_type": <type of event>,
  "events": [
    <N-times the basic message format>
  ]
}
```

- *event_count* contains the number of events listed in *events* property.
- *event_type* indicates the type of events listed in *events* property.
- *events* lists the event data, with same event type for all list items.

6.3 Basic event structure

Each event contains of a set of properties:

```
{  
  "event_id"       : <unique numeric value to identify event>,  
  "event_timestamp" : <timestamp of event in UTC>,  
  "event_data"      : <event-specific data>  
}
```

- *event_id* is a unique numeric value. This can help to identify event data, that has already been processed.
- *event_timestamp* shows, when the event occurred. This is not the timestamp, when the notification was sent.
- *event_data* contains the event-specific data.

6.4 Examples

Here are some examples to show complete JSON structures.

6.4.1 mailing_opened (one event)

This example shows the data for the *mailing_opened* event containing data for one event:

```
{
  "event_count": 1,
  "event_type": "mailing_opened",
  "events": [
    {
      "event_id": 12345678,
      "event_timestamp": "2021-02-18T13:06:45Z",
      "event_data": {
        "mailing_id": 123456,
        "recipient_id": 4567,
        "recipient_data": {
          "some_profile_field": "value1",
          "other_profile_field": "123"
        }
      }
    }
  ]
}
```

6.4.2 mailing_opened (multiple events)

This example shows the data for the *mailing_opened* event containing data for multiple events:

```
{
  "event_count": 3,
  "event_type": "mailing_opened",
  "events": [
    {
      "event_id": 12345678,
      "event_timestamp": "2021-02-18T13:06:45Z",
      "event_data": {
        "mailing_id": 123456,
        "recipient_id": 4567,
        "recipient_data": {
          "some_profile_field": "value1",
          "other_profile_field": "123"
        }
      }
    },
    {
      "event_id": 12345689,
      "event_timestamp": "2021-02-18T13:06:47Z",
      "event_data": {
        "mailing_id": 123462,
        "recipient_id": "not_tracked",
        "recipient_data": {}
      }
    },
    {
      "event_id": 12345722,
      "event_timestamp": "2021-02-18T13:07:02Z",
      "event_data": {
        "mailing_id": 123210,
        "recipient_id": 3245,
        "recipient_data": {
          "some_profile_field": "value2",
          "other_profile_field": "456"
        }
      }
    }
  ]
}
```

The recipient in the second event data has rejected to be tracked. *recipient_id* shows "not_tracked" and the profile field data is empty.

6.4.3 link_clicked

This example shows the data for the *link_clicked* event containing data for one event:

```
{
  "event_count": 1,
  "event_type": "link_clicked",
  "events": [
    {
      "event_id": 12345690,
      "event_timestamp": "2021-02-18T13:08:21Z",
      "event_data": {
        "mailing_id": 123459,
        "recipient_id": 5678,
        "recipient_data": {
          "some_profile_field": "value1",
          "other_profile_field": "123"
        },
        "link_id": 23456789012
      }
    }
  ]
}
```

7 Events

7.1 Mailing delivery

7.1.1 mailing_delivered

Description

The *mailing_delivered* event occurs, when a mailing has been successfully delivered to a recipient. The event is not triggered for admin and test mailings.

Please note: Delivery of a mailing is determined by the response of the mail provider. It depends on the mail provider, if the mailing appears in the recipients inbox.

JSON data

```
{
  "mailing_id"      : <ID of mailing>,
  "recipient_id"    : <ID of recipient>,
  "recipient_data" : {
    <selected profile field data>
  }
}
```

- *mailing_id* denotes the ID of the mailing
- *recipient_id* denotes the ID of the recipient, who received the mailing
- *recipient_data* contains a set of selected profile fields and its content

7.1.2 hard_bounce

Description

The event *hard_bounce* indicates, that the mailing could not be sent to the recipient due to a permanent error (like unknown or invalid mail address).

JSON data

```
{
  "mailing_id"      : <ID of mailing>,
  "recipient_id"    : <ID of recipient>,
  "recipient_data" : {
    <selected profile field data>
  }
}
```

- *mailing_id* denotes the ID of the mailing
- *recipient_id* denotes the ID of the recipient
- *recipient_data* contains a set of selected profile fields and its content

7.1.3 mailing_delivery_complete

Description

This event indicates, that the mailing has been sent to all recipients and each mailing has either been delivered to its recipients or the delivery lead to a hard bounce (see [hard_bounce](#) event).

JSON data

```
{
  "mailing_id"      : <ID of mailing>
}
```

- *mailing_id* denotes the ID of the mailing

7.2 Recipient reactions

7.2.1 link_clicked

Description

The *link_clicked* event occurs, when a recipients clicks on a link in a mailing.

Please note: Link click detection is only possible, if the link is trackable.

JSON data

```
{
  "mailing_id"      : <ID of mailing>,
  "recipient_id"    : <ID of recipient>,
  "recipient_data" : {
    <selected profile field data>
  },
  "link_id"         : <ID of clicked link>
}
```

- *mailing_id* denotes the ID of the mailing
- *recipient_id* denotes the ID of the recipient, who opened the mailing
- *recipient_data* contains a set of selected profile fields and its content
- *link_id* denoted the ID of the clicked link

7.2.2 mailing_opened

Description

The *mailing_opened* event indicates, that a recipient has opened the mailing in a mail client.

Please note: Measuring opening a mailing is only possible, if the mailing contains a tracking pixel.

JSON data

```
{
  "mailing_id"      : <ID of mailing>,
  "recipient_id"    : <ID of recipient>,
  "recipient_data" : {
    <selected profile field data>
  }
}
```

- *mailing_id* denotes the ID of the mailing
- *recipient_id* denotes the ID of the recipient, who opened the mailing
- *recipient_data* contains a set of selected profile fields and its content

7.2.3 binding_changed

Description

The *binding_changed* event occurs, when the status of a recipient binding has changed.

Please note:

- No event is fired on imports or auto imports.
- In some cases, the property "mediatype" is not present or set to null.

JSON data

```
{
  "mailinglist_id" : <ID of mailinglist>,
  "mediatype"      : <media type>,
  "recipient_id"    : <ID of recipient>,
  "status"          : <new status>,
  "opt_out_by"      : <opt out by>,          // Only present for status "opt_out"
  "recipient_data" : {
    <selected profile field data>
  }
}
```

- *mailinglist_id* denotes the ID of the mailing list
- *mediatype* denotes the media type (see [Media types](#) for list of valid values). May not present in some cases.
- *recipient_id* denotes the ID of the recipient
- *status* describes the user status (see [User states](#) for list of valid values)
- *opt_out_by* describes who opt-out the recipient (only present if *status* is "opt_out", values can be "recipient" or "admin")
- *recipient_data* contains a set of selected profile fields and its content.

7.2.4 profile_field_changed

Description

The *profile_field_changed* event occurs, when the value of a recipient profile field has been changed.

Please note:

- To be able to report a changed value of a profile field, this profile field has to be set to "include profile field in history".

JSON data

```
{
  "recipient_data": {
    "email"      : <email address>,
    "gender"     : <type of gender>,
    "firstname"  : <firstname>,
    "lastname"   : <lastname>,
    "mailtype"   : <mailtype>,
    "sys_tracking_veto" : <veto flag>,
    <selected profile field data>
  },
  "recipient_id" : <ID of recipient>
}
```

8 Tables

8.1 User states

User states represents the state of recipients on a mailing list.

Value	Description
active	Recipient is active and can receive mailings
bounce	Recipient marked as bounced will not be delivered due to hard bounce (like unknown email address) or too many soft bounces (like capacity of inbox exceeded)
opt_out	Recipient has been unsubscribed (either recipient unsubscribed itself or has been unsubscribed by an EMM user)
wait_for_confirm	User has been subscribed, but not confirmed subscription.
blacklisted	User has been blacklisted.
suspended	User has been temporarily disabled.

8.2 Media types

Value	Description
email	Email newsletter
sms	SMS
post	Mail

9 Changelog

Version	Date	Edited by	Description
1.0.0	2021-03-29	Markus Dörschmidt	Initial version
1.0.1	2021-07-28	Markus Dörschmidt	Fixed typing errors
1.1.0	2023-01-09	Markus Dörschmidt	Added binding_changed , added section Tables , fixed typing errors
1.1.1	2023-09-07	Markus Dörschmidt	Fixed minor typos
1.1.2	2023-09-25	Markus Dörschmidt	Fixed minor typos
1.1.3	2025-05-30	Eva Herrmann	New formatting and design adjustments, addition to 7.1.1
1.1.4	2025-09-10	Eva Herrmann	Added chapter 7.2.4 <code>profile_field_changed</code>

