



# Engage Delivery Cloud

---

PRODUCT MANUAL | August 2023

# 1 Foreword

## Copyright

The contents of this manual cover material copyrighted by Marigold. Marigold reserves all intellectual property rights on the manual, which should be treated as confidential information as defined under the agreed upon software licence/lease terms and conditions.

The use and distribution of this manual is strictly limited to authorised users of the Marigold Interactive Marketing Software (hereafter the “Software”) and can only be used for the purpose of using the Software under the agreed upon software licence/lease terms and conditions. Upon termination of the right to use the Software, this manual and any copies made must either be returned to Marigold or be destroyed, at the latest two weeks after the right to use the Software has ended.

With the exception of the first sentence of the previous paragraph, no part of this manual may be reprinted or reproduced or distributed or utilised in any form or by any electronic, mechanical or other means, not known or hereafter invented, included photocopying and recording, or in any information storage or retrieval or distribution system, without the prior permission in writing from Marigold.

Marigold will not be responsible or liable for any accidental or inevitable damage that may result from unauthorised access or modifications.

User is aware that this manual may contain errors or inaccuracies and that it may be revised without advance notice. This manual is updated frequently.

Marigold welcomes any recommendations or suggestions regarding the manual, as it helps to continuously improve the quality of our products and manuals.

## 2 Table of Contents

<b>1</b>	<b>Foreword</b>	<b>2</b>
<b>2</b>	<b>Table of Contents</b>	<b>3</b>
<b>3</b>	<b>Goal of this document</b>	<b>4</b>
<b>4</b>	<b>Prerequisites</b>	<b>5</b>
<b>5</b>	<b>Engage Delivery Cloud</b>	<b>6</b>
5.1	Available endpoints	6
<b>6</b>	<b>Authentication</b>	<b>7</b>
6.1	Basic Authentication	7
6.2	JWT authentication	7
6.3	Which authentication should I use?	7
<b>7</b>	<b>Sending email messages</b>	<b>8</b>
7.1	Structure of the /send request	8
7.2	Example of the /send request	10
7.3	Response of the /send request	11
7.4	Example of the /send response	11
<b>8</b>	<b>Processing feedback</b>	<b>12</b>
8.1	Example of the feedback	12
<b>9</b>	<b>Security measures</b>	<b>13</b>

### 3 Goal of this document

This document contains the technical documentation of Engage Delivery Cloud. After reading this document, the reader should be able to perform valid HTTPS calls to the API and interpret the results. This is a technical document: it will in no way provide a guide on how to create valid email content.

## 4 Prerequisites

Before using Engage Delivery Cloud, Marigold will configure an *Account* and one or more *Mail domains* for the customer. This ensures that only users that are authorized to use the API can send out emails, and that customers can send emails from their mail domains only.

Whenever a user tries to send an email from an unknown mail domain or a mail domain that isn't owned by them, the mail will not be sent out.

In order to set up your account and/or mail domain(s), contact Marigold support.

## 5 Engage Delivery Cloud

Engage Delivery Cloud is a web service that enables an authenticated user to send out email messages via the GRID platform.

### 5.1 AVAILABLE ENDPOINTS

ENDPOINT	HTTP VERB	FUNCTION	NEEDS AUTHENTICATION
/email/send	POST	Send out email messages	Yes
/authentication/authenticate	POST	Obtain a JWT token	Yes

## 6 Authentication

Engage Delivery Cloud supports 2 kinds of authentication: Basic Authentication, and authentication via JSON Web Tokens (JWT)<sup>1</sup>.

### 6.1 BASIC AUTHENTICATION

When using basic authentication, **each** call to the API needs to have an Authorization header in the following format:

KEY	VALUE
Authorization	Basic Base64Encode(<username>:<passcode>)

For example; if the username is *username*, and the passcode is *notsosecurepasscode*, the header would look like this:

KEY	VALUE
Authorization	Basic dXNlcm5hbWU6bm90c29zZWVwYXNzY29kZQ==

The username and passcode will be provided by Marigold when subscribing to the platform.

### 6.2 JWT AUTHENTICATION

When using JWT authentication, **each** call needs to have a *Bearer Token*. This bearer token can be obtained by doing a call to the *Authenticate* endpoint. A token will expire after 24 hours. When this token has been expired, a new token can be obtained using the *Authenticate* call.

Calls to the *Authenticate* endpoint also require authentication. Both Basic and JWT are supported.

Example of the authorization header with the bearer token:

KEY	VALUE
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp... <snipped>...-gAWQT8l

### 6.3 WHICH AUTHENTICATION SHOULD I USE?

JWT is the preferred form of authentication.

---

<sup>1</sup> <https://jwt.io/introduction/>

## 7 Sending email messages

### 7.1 STRUCTURE OF THE /SEND REQUEST

The /send endpoint expects a POST call with a JSON payload, that consists of an array of email objects. The structure of these objects is listed below. Each POST request can contain up to 100 messages.

Note that you can only send emails from a domain that is registered by Marigold and linked to the authenticated account.

#### Email object

PROPERTY	DATA TYPE	VALUE	REQUIRED
reference	String	A unique identifier for the email message	yes
content	Content object	The content of the email. <a href="#">See below</a>	yes
headers	Headers Object	The headers of the email. <a href="#">See below</a>	yes
tags	Array of strings	Tags that can be sent along for reporting uses	no
customsendtime	DateTime	The time (within the next 24h) when you want the email to be scheduled	no

#### Content object

PROPERTY	DATA TYPE	VALUE	REQUIRED
html	String	The Html content of the email	yes
text	String	The text content of the email	yes
attachments	Array of Attachment objects	Email attachments. Either inline or with an URI to a remote attachment. <a href="#">See below</a>	no

#### Attachment object

PROPERTY	DATA TYPE	VALUE	REQUIRED
filename	String	The name of the attachment	yes
mimetype	String	The Mime Type of the attachment	yes
data	String	The base64-encoded data	no*
uri	String	The Uri of the remote attachment	no*

\* either 'data' or 'uri' is required.



### Headers object

PROPERTY	DATA TYPE	VALUE	REQUIRED
subject	String	The subject of the email	yes
to	EmailEntity	The Recipient. <a href="#">See below</a>	yes
from	EmailEntity	The Sender. <a href="#">See below</a>	yes
reply	EmailEntity	Reply Address. <a href="#">See below</a>	yes
listunsubscribe	String	The List-Unsubscribe header	yes
optional	Array of Header objects	An array of headers that will be added to the message. <a href="#">See below</a>	no

### EmailEntity object

PROPERTY	DATA TYPE	VALUE	REQUIRED
address	String	The email address of the entity	yes
alias	String	The alias of the entity	no

NOTE: The address of the sender (from header) MUST use a mail domain that is configured for your account. Example: if you own the domain *exampledomain.com*, a valid address will be 'newsletter@exampledomain.com'.

### Header object

PROPERTY	DATA TYPE	VALUE	REQUIRED
name	String	The header name (MUST start with 'X-')	yes
value	String	The header value	yes

## 7.2 EXAMPLE OF THE /SEND REQUEST

Below you can find an example of a Send call:

```
[
  {
    "reference": "2020-02-01-01",
    "content": {
      "html": "This is the HTML content of the message",
      "text": "This is the Text content",
      "attachments": [
        {
          "filename": "Placeholder.jpg",
          "uri": "https://via.placeholder.com/300",
          "mimeType": "image/jpg"
        }
      ]
    },
    "headers": {
      "subject": "Example of an email",
      "to": {
        "Alias": "John Doe",
        "address": "johndoe@selligent.com"
      },
      "from": {
        "Alias": "Email Sender",
        // The address MUST use a domain that is owned by the sender
        "Address": "sender@senderdomain.slgnt.eu"
      },
      "reply": {
        "Alias": "Reply",
        "Address": "reply@senderdomain.com"
      },
      "listunsubscribe": "<http://www.example.com/unsubscribe.aspx?ID=123456>
, <mailto:unsubscribe@example.com?subject=unsubscribe 123456>",
      "optional": [
        {
          "name": "x-optionalHeader",
          "value": "headervalue"
        }
      ]
    },
    "tags": [
      "ExampleMail", "Documentation"
    ]
  }
]
```

## 7.3 RESPONSE OF THE /SEND REQUEST

Whenever a call has been successful, the response of the POST request will contain an array of result objects; for each message in the request, a result object will be returned:

Result object

PROPERTY	DATA TYPE	VALUE	REQUIRED
reference	String	The reference that was used in the request	reference
gridid	String	A unique identifier that is used as an identifier by the platform	gridid

These results can be used to map a *reference* (client side identifier of a message) to the *gridid* (platform side identifier of a message) when processing the feedback.

## 7.4 EXAMPLE OF THE /SEND RESPONSE

And a corresponding response:

```
[
  {
    "reference": "2020-02-01-01",
    "gridid": "40775ca7-8af1-457d-bf96-42bb8c6716ed"
  }
]
```

## 8 Processing feedback

When an email has been processed by the system (either successfully delivered, or bounced), a feedback object will be sent to a webhook.

A client can hook to this webhook to process the results of the emails. Each call will contain an array of Feedback objects. Each object will contain information about the message:

Feedback object

PROPERTY	DATA TYPE	VALUE
accountid	String	The account that was used to send out the message
clientid	String	The client that was used to send out the message
gridid	String	The platform identifier for the message. This maps to the <i>GridId</i> property of the result
smtpoutcome	String	The smtp response of the mail server/the reason why a message could not be sent
sendtime	DateTime	When the message was sent
result	Integer	A status code representing the result of an email

The explanation of the status codes is listed below:

CODE	MEANING
1	Success
2	Undeliverable

### 8.1 EXAMPLE OF THE FEEDBACK

A webhook will receive the following POST body:

```
{
  "accountId": 5,
  "clientId": 3,
  "gridId": "40775ca7-8af1-457d-bf96-42bb8c6716ed ",
  "smtpOutcome": "250 2.6.0 15008 bytes received in 00:00:00; Message id 20200213
0526509467 accepted for delivery",
  "sendTime": "2020-02-13T05:26:51.473+00:00",
  "result": 1
}
```

## 9 Security measures

All calls to the API require HTTPS / TLS1.2 transport encryption. All content data is stored for a maximum of 48 hours using AES256 encryption. (FIPS 140-2 compliant)