

# sinch



## **Migration guide from E365 to Mailgun**

Version 1 – March 2023

[sinch.com](https://sinch.com)



# Revision History

Version	Date	Description
1.0	23 Mar 23	Baseline draft



# Table of Contents

<b>1 Document purpose</b>	<b>3</b>
<b>2 Initial Setup</b>	<b>4</b>
2.1 Creating an account	4
2.2 Adding domain/s	4
<b>3 Authorization and security</b>	<b>5</b>
3.1 Base URL	5
<b>4 Sending a message – E365 VS Mailgun</b>	<b>6</b>
4.1 Request / message send API parameters	6
4.2 Response sending a message	10
<b>5 Retrieving message status/events</b>	<b>12</b>
5.1 Retrieving specific event types	13
<b>6 Suppressions</b>	<b>15</b>



# 1 Document purpose

This migration guide is intended to provide high level overview and basic understanding of the main differences between Email 365 and Mailgun in order to estimate the work required to switch from one platform to the other.

Being the leading developer-oriented platform, Mailgun has a highly detailed online documentation with in-depth examples and explanation of each API endpoint and setting, available at <http://documentation.mailgun.com> with code samples in the most popular programming languages.

It is worth mentioning that both platforms allow for sending an email with text or HTML parts, adding attachments and getting information back from the API about what happened to the message. Both platforms use RESTful APIs with JSON responses and the ability to use SMTP relay in addition to sending MIME messages in a similar fashion. Mailgun is a much more robust platform, with significantly more ways to configure sending, schedule messages, retrieve stats, receive messages, store templates, verify addresses and much more.



## 2 Initial Setup

### 2.1 Creating an account

Mailgun is entirely cloud based and initial setup must be performed after signing-up to the service at <https://signup.mailgun.com/new/signup> compared to Email 365 where you will get this provisioned after a formal request to the team.

Once the account is created it will be possible to send messages to previously validated email addresses from a sandbox domain. In order to send with your own domain a trial period can be enabled if you are eligible, or a paid plan must be purchased either as self-service or a contract.

### 2.2 Adding domain/s

While in Email365 you are must request a new domain to be added for you and wait for the process to complete, it is entirely possible to create, validate and start sending from thousands of domains without waiting for provisioning and you have the ability to automate the creation of new domains using the Rest API.

Before being able to send with your domain/s you need to validate them by setting up SPF and DKIM authentication. At this point it is advisable to also setup the rest of the DNS records like MX and CNAME that allow inbound emails, click tracking, etc.

To read more about the process of verifying a domain, please refer to

[https://documentation.mailgun.com/en/latest/user\\_manual.html#verifying-your-domain-1](https://documentation.mailgun.com/en/latest/user_manual.html#verifying-your-domain-1) . The same article contains more information on DNS records and links to common DNS providers to help with the setup.

Once SPF and DKIM are added, you will be allowed to send messages from this domain.

You can add multiple domains and it is advisable to use different subdomains to separate mail traffic depending on its content and type – for example marketing from transactional type emails.



## 3 Authorization and security

When migrating between API or SMTP from E365 to Mailgun you will need to generate new credentials to authorize your calls into Mailgun, but the type of authorization remains the same – basic, requiring a username and a password. The username will always be the same - “api”, but the password will differ from domain-to-domain and can be regenerated if needed. Compared to E365, you don’t need to contact support in order to generate new credentials, it is fully self-managed. Additionally, it is possible to generate multiple send-only credentials for different sub-systems or purposes, allowing you to revoke each separately if needed. Here is an article that provides more details on authorization:

<https://documentation.mailgun.com/en/latest/api-intro.html#authentication-1>

When it comes to security both platforms offer TLS encryption of both HTTP or SMTP connections with Mailgun allowing you to tweak security to a much deeper extent, such as enforcing TLS upon delivery, using SSO when signing-in to the front-end and more. You can read more about TLS enforcing here

[https://documentation.mailgun.com/en/latest/user\\_manual.html#tls-sending-connection-settings](https://documentation.mailgun.com/en/latest/user_manual.html#tls-sending-connection-settings) and more about SAML SSO here

<https://help.mailgun.com/hc/en-us/articles/1260801912270-SAML-SSO>

### 3.1 Base URL

When making API calls to send messages through E365 your base URL will remain the same irrespective of the domain your messages are sent from. With Mailgun the base URL will change depending on the data region you have selected for the specific domain, thus it is important to take this into consideration when creating your integration. Additionally, most API URLs must include the domain. Here is a document giving more information on that topic

<https://documentation.mailgun.com/en/latest/api-intro.html?highlight=base%20url#base-url-1>



## 4 Sending a message – E365 VS Mailgun

Sending a message on Mailgun is simpler compared to doing the same over Email365. The reason is primarily the functionality of Mailgun to assemble a MIME message out of the most basic information provided by the sending application, without the requirement to define traits such as character encoding or content encoding. At the same time Mailgun allows you to compose a MIME message with a library entirely of your choice, meaning that you can mimic the API calls you have been making with Email 365 so far to a very large extent, minimizing the time required to reform the requests on your side.

### 4.1 Request / message send API parameters

Sending messages over API for both products is simple, but essentially the products are different and because of this the names of the fields of the request are different too, to make it easier to switch, we have outlined the difference below:

		Required		Note
Email 365	Mailgun	Email 365	Mailgun	
notificationId	None	No	No	Just as in Email 365, Mailgun will respond with a unique message ID after a message is accepted for processing over SMTP or API. On both platforms this is an ID to use to filter the API rather than used to send a message as the ID is defined by the system.
sender	from	No	Yes	While a default from address can be defined in Email 365 this is not possible in Mailgun and thus has to be defined for each message/group of messages.



recipients	to		Yes	Yes	Mailgun allows for up to 1000 recipients to be listed in a single To field or multiple To fields to be used to declare batch recipients.
contentText	text	html	No*	Yes*	This is required on E365 if there are no attachments, and it is required on Mailgun if a template is not being used. You can choose either text or html or both, depending on your use-case.
contentTextType	None		Yes	No	Mailgun does not need this declaration as it will automatically pick up on the type of content based on the text/html fields used or if it is a template.
contentTextEncoding	None		Yes	No	Mailgun will automatically assign this upon sending, so the field might only be used if you are building your own mime message
characterSet	None		No	No	Mailgun will set UTF-8 by default, same as E365, but if you are building your own MIME message you can define that depending on your library used.
contentAttachments	attachment		No	No	Mailgun allows you to attach files to your messages without prior approval, compared to Email 365 where you need prior configuration. The limit is 25mb per message for all attachments combined.
contentName	None		No	No	Mailgun will automatically pick up the name of the attachment from the file you have selected.
contentId	None		No	No	Mailgun will use the name of the file, but if an ID is required you can post a MIME of your choice that contains this.
attachment	attachment		No	No	In Mailgun the attachments' content is defined with the declaration of each attachment, therefore it is not necessary to separate the names and list of attachments from the content.





contentType	None	No	No	Mailgun will automatically pick up the type of the attachment and will declare it in the MIME message, therefore it is not necessary to be included with your call.
channelPreferences	None	No	No	Mailgun allows only the sending of email therefore a declaration of the channel is not relevant.
systemId	o:tag	No	No	Mailgun allows for tagging a single message with up to three tags, which can then be used to filter the API and achieve statistical aggregation. Therefore, replacing the systemId of Email 365 with Tagging in Mailgun.
campaignId	o:tag	No	No	Mailgun allows for tagging a single message with up to three tags, which can then be used to filter the API and achieve statistical aggregation. Therefore, replacing the systemId of Email 365 with Tagging in Mailgun.
processingType	None	No	No	Mailgun has automatic queue management that does not need declaration for processing type.
expiryHour	None	No	No	Mailgun will retry a message for 8 hours after the first attempt. This time period cannot be changed.
email	None	No	No	As Mailgun is sending only email an email-specific configuration is not required and all settings defined under this object in Email 365 are done on general level in Mailgun.



subject	subject	Yes	Yes	A subject is always required on both platforms.
cc	cc	No	No	Using the CC field is optional for both platforms.
bcc	bcc	No	No	Using the CC field is optional for both platforms.
senderName	None	No	No	In Mailgun defining the “good name” of the sender is done within the “from” field, as such:  Sender’s Name <YOU@YOUR_DOMAIN_NAME> where “Sender’s Name” would be the from name.
replyTo	h:Reply-To	No	No	In both platforms defining a reply-to address is optional.
replyToName	None	No	No	In Mailgun defining the “good name” of the reply-to address is done within the “h:Reply-To” field, as such:  Reply to Name <YOU@YOUR_DOMAIN_NAME> where “Reply to Name” would be the reply-to name.
alternateContentText	text   html	No	No	Depending on whether text or html was used for the message already, you can choose an alternate type of content.
antivirusScan	None	No	No	Mailgun does not offer antivirus scan on inbound or outbound mail, compared to Email 365 which does.
alternateContentText Type	None	No	No	Mailgun will automatically assign the content type depending on the content type chosen (text and/or html)
alternateContentTextEncoding	None	No	No	Mailgun will not encode automatically, equivalent to



					what Email 365 does when this is not declared, but if you need to you can construct and send an entire MIME message with a library of your choice.
deliveryU TCDate	deliveryU TCHour	o:deliverytime	No	No	Mailgun also allows scheduling of messages, but from 2 to 72 hours after the initial call was made. Compared to Email 365 where the Date and Hour are in separate fields, Mailgun combines them into one.
headers		message-headers	No	No	Mailgun allows you to add a list of all MIME headers dumped to a json string, same as Email 365.

For a detailed list of Send message parameters and examples in different programming languages, please refer to <https://documentation.mailgun.com/en/latest/api-sending.html#sending> and [https://documentation.mailgun.com/en/latest/user\\_manual.html#sending-via-api](https://documentation.mailgun.com/en/latest/user_manual.html#sending-via-api)

## 4.2 Response sending a message

There is a significant and at the same time very simple difference between the response that Mailgun gives when sending a message and the one that Email 365 gives – while Email 365 will respond with account ID, notification ID, timestamp, recipient, charset, recipients, headers and a lot more, upon a successfully accepted message the Mailgun API will respond only with the ID that you can use to later filter the API.



Example:

```
{  
  "id": "<20240721132456.0dcce6421nnac1871@sender.domain>",  
  "message": "Queued. Thank you."  
}
```

In case of an error, you will receive the respective response back. For a list of error codes, please refer to <https://documentation.mailgun.com/en/latest/api-sending.html#error-codes>



## 5 Retrieving message status/events

Filtering the events (such as accepted, delivered, clicked, opened, bounced, etc.) stored in database is possible on both platforms where Mailgun offers much more filtering options and forming the responses. The log retention is 7 days for Email 365 and up to 30 days for Mailgun for paid plans.

Both platforms rely on a GET call and on the URL itself to filter the database for historical information on the sent messages and both can use the ID of the message that is returned by the API after a message is sent. Here is a comparison based on the Email 365 APIs abilities to filter for message stats:

Field name		Required		Note
Email 365	Mailgun	Email 365	Mailgun	
accountId	tags	Yes	No	While the accountId is required in Email 365 in order to get any results back, on Mailgun it is optional and depends on the desired output. However, if you choose to replace the accountId with a custom tag of your choice you can do so and filter the API with it.
notificationId	message-id	Yes	No	The notificationId in Email 365, or the message-id in Mailgun are both returned in the response of the POST call when sending a message. The difference is that on Mailgun it is not required to filter.
recipient	to	No	No	Both platforms allow for filtering of events based on the recipient with a slightly different syntax between them. Please refer to the examples below for more information.



Here is a comparison between the URLs to achieve the same on the platforms:

Query status of a specific message (GET)	
Email 365	Mailgun
<code>https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/id/10000000902077</code>	<code>https://api.mailgun.net/v3/{{yourdomain}}/events?message-id:20250421155428.5277ee3e52435cb3@your.domain</code>

In the example above Email 365 is using the notification ID (10000000902077), provided as a response to the POST call when sending the message, in order to get back a result from the API as to what happened to the message. Mailgun’s example is the same, but instead of the account name you need the domain and same as in Email 365 – the ID that was returned when sending the message (20250421155428.5277ee3e52435cb3@your.domain).

Query status of a specific message and specific recipient of that message (GET)	
Email 365	Mailgun
<code>https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/id/10000000902077?recipient=email:testuser@sinch.com</code>	<code>https://api.mailgun.net/v3/{{yourdomain}}/events?message-id:20250421155428.5277ee3e52435cb3@your.domain&amp;to=testuser@sinch.com</code>

In the example above the same is achieved on both platforms – filtering for a message (with its ID), and then further filtering for events related to that recipient only.

## 5.1 Retrieving specific event types

In Mailgun several different events can be generated as a result of sending a message, depending on your settings (e.g., is open tracking enabled) and on what happened to the message afterwards. The list of events can be found here:

<https://documentation.mailgun.com/en/latest/api-events.html#event-types>



In Mailgun you can ask for bounce events to be pulled for a specific period, just the same as in Email 365, here is a comparison of the GET API calls:

Query bounce events for a specific time-frame (GET)	
Email 365	Mailgun
https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/email/bounce?status Code=CH_REJECTED&startUTCTime=20160727115959&endUTCTime=20160805115959	https://api.mailgun.net/v3/{yourdomain}/events? event=failed&begin=1640995200&end=1641081600

The way to achieve these results is pretty much the same in Mailgun - “/bounce” is replaced by “event=failed” and “startUTCTime”/“endUTCTime” is replaced by “begin”/“end”, with the difference that you can use Epoch time in seconds, and you can also filter many other traits of the message such as attachments, subject, size, etc. For more information on filtering events, please visit this documentation page: <https://documentation.mailgun.com/en/latest/api-events.html#filter-field>

In order to retrieve all other types of events we store, you must replace “failed” in the above examples with one from the list found here:

<https://documentation.mailgun.com/en/latest/api-events.html#event-types>



## 6 Suppressions

Both Email 365 and Mailgun will suppress sending to addresses that have previously bounced with such status, that was deemed permanent, the so called “hard bounces”. While Email 365 has an explicit list of exceptions from the general rule that hard bounces are generated when an SMTP status code starting with 5.x.x. occurs, Mailgun will generally treat all SMTP status codes as soft-bounce and will not suppress sending in order to avoid preventing mail from going out without reason. However, based on Mailgun’s year-long experience certain status codes will be treated as hard in order to spare reputation damage from incurring on the sender. When a suppression occurs in Email 365, the next attempt to send to the same recipient may be stopped if it happens within 7 days of the initial message that bounced, however on Mailgun once suppressed, a recipient will stay in the suppression list until the admin of the Mailgun account removes it. The removal or addition of an email address from/to the suppression list can be done both via the front-end and via the API, allowing for complete automation and potential mirroring of the functionality that Email 365 presently delivers.

Additionally, Mailgun will suppress sending to recipients who have reported a message as spam or have chosen to unsubscribe. The same rule applies here too – the owner of the Mailgun account which was used to send the first bounced message can remove the recipient from the list manually or automatically by using the API.

You can read more about suppressions here

<https://documentation.mailgun.com/en/latest/api-suppressions.html>