Use cases and recommended methods

This page contains many collapsed sections that contain details about recommended methods in REST, HTTP GET/POST API. Use the "Click to expand" links to reach hidden parts.

On this page:

- Manage single recipients (subscribe, unsubscribe, update)
- Synchronize recipients between your application and MailUp (i.e. bulk import and/or export)
- Use confirmed opt-in (COI)
- Manage suppression lists (i.e. "do not email" lists of recipients)
- Send transactional and other personalized emails
- Bulk email messages
- Send Transactional or Bulk SMS
- Extract or display statistics

Manage single recipients (subscribe, unsubscribe, update)

MailUp provides several ways to add a new subscriber and manage either data fields or subscription status. If you're in the process of developing a generic application interacting with multiple accounts, it must be considered that personal data fields may differ from account to account.

We recommend to always make use of the "retrieve field names" feature, returning the actual field names and Id occurences for a particular admin console account. Those returned occurrences can be used to update personal data recipient fields afterwards.

HTTP GET/POST and subscription forms

Subscription forms are based on HTTP GET/POST methods, which allow subscription unsubscription, check subscription status and update for a single recipient. These methods support optin, confirmed optin, optout, fields update, unsubscription. Refer to this page if you want to build a custom subscription form using AJAX. Retrieve fields feature is not available with HTTP GET/POST.

Action	Comments	Get/Post API
Confirm successful communication with the API	This method may be useful if your application needs to ping mailup to check current authentication and connection statuses.	WSActivation.aspx
Activation	You can enable HTTP GET/POST calls for any caller, you can restrict usage to a set of IPs or you can call a method that registers the caller IP in order to add it to the list of authorized IPs.	WSActivation.aspx
Single Subscribe (client side)	Subscribes a single recipient to one or more lists (e.g. during checkout or registration) by calling a page that redirects to a landing page displaying the result	Subscribe.aspx
Single Subscribe (server side[1])	Subscribes a single recipient to one or more lists (e.g. during checkout or registration) and returning a result to the caller	Xmlsubscribe.aspx
Single Unsubscribe (server side[1])	Unsubscribes a single recipient from one or more lists (e.g. customer updates their account in the storefront) and returning a result	XmlUnsubscribe.aspx
Single Update (server side[1])	Updates a single recipient based on mapped data fields (e.g. first name, last name, etc) and returning a result	XmlUpdSubscriber.aspx
Retrieve subscription status for a specific customer and list (server side[1])	Retrieves recipient's status (subscribed, pending or unsubscribed)	XmlChkSubscriber.aspx

(1) requires server side development, and in case of AJAX implementations, this operation cannot be done directly from a client page because cross domain requests are blocked by the "same origin policy"

REST API

Rest API handles subscription status, group membership and personal data fields. In addition, it is the only MailUp solution that allows removal of a single recipient from a group. Adding a new recipient is an asynchronous process (i.e. the method's response does not mean that the operation is completed, you could use a specific method to get the status of requested operation). Retrieve fields feature is available with GET /Console /Recipient/DynamicFields

Although the MailUp REST API is the most complete solution for managing a single subscriber, please keep in mind that it is subject to rate limiting, so some calls could be blocked if you are exceeding the allowed number of calls/s. For this reason, if your source is a web form with high subscription traffic you would be better off in considering the HTTP GET/POST for new subscriptions.

Action	Comments	REST API
Confirm successful communication with the API	Returns authentication information.	GET /Console/Authentication/Info
Retrieve Lists	Retrieves all lists related information	GET /Console/User/Lists
Retrieve Groups	Retrieves groups for a certain list	GET /Console/List/{ID_LIST}/Groups
Retrieve Fields	Retrieves the actual personal data field names along with ids. Useful for adding or updating subscribers' personal information afterwards.	GET /Console/Recipient /DynamicFields
Single Subscribe	Subscribes a single recipient to one lists (e.g. during checkout or registration). Confirmed optin process can be implemented by means of ConfirmEmail parameter	POST Console/List/{ListID}/Recipient (without COI) POST Console/List/{ListID} /Recipient?ConfirmEmail=true (with COI)
Single Unsubscribe	Unsubscribes a single recipient from one or more lists (e.g. customer updates their account in the storefront)	DELETE /Console/List/{id_List} /Subscribe/{id_Recipient}
Single Update	Updates a single recipient based on mapped data fields (e.g. first name, last name, etc)	PUT /Console/Recipient/Detail
Retrieve subscription status for a specific customer and list	Returns all subscribed recipients to a certain list	GET /Console/List/{id_List} /Recipients/Subscribed?filterby=" Email.Contains('{Email}')"

Webhooks

You can configure webhooks to enable a fast callback to your website everytime a recipient changes the subscription status or any of the personal data fields (e.g. by filling in and submitting to an autoprofile form). Callback method can be configured in order to include recipient's details as method parameter. Webhooks are managed by a worker process that is like a "frequent cron": you may experience some minutes of delay between the action and the corresponding callback. Bounces cannot be notified through webhooks.

Synchronize recipients between your application and MailUp (i.e. bulk import and/or export)

You may need to transfer a set of recipients, possibly with their personal data, from your application to MailUp; or viceversa. Solutions that work well with a single recipient may not be suitable for big amounts of recipients, in this case bulk solutions are required.

When planning a synchronization approach you must take care about how unsubscription is made. In particular, you should be aware that, when importing subscribers that are already present and unsubscribed in specified MailUp list, unsubscription status is not changed, except for the cases in which you deliberately set an API option that forces subscription.

- If unsubscription is natively managed by MailUp via the "Unsubscribe" link normally nested in the email footer, forcing subscription should be considered only when an unsubscribed recipient specificly asks to be subscribed again
- If unsubscription is managed by an external system outside MailUp (i.e the "Unsubscribe" link in email footer leads to an external site) then the external application runs as "master" database and you should consider to force subscription whenever you import recipients.

Several possible solutions are available.

FTP based integrations

You can request your MailUp representative for a FTP Batch Process for massive data transfer using files CSV files. This feature can be used for very large amounts of recipients (this kind of process has been successfully tested with 4 millions of recipients in a single file).

You can request this feature to

- Import recipients as subscribed
- Import recipients as unsubscribed
- Export unsubscribed
- Export subscribers
- Export statistics
- Export bounces

Data transmission does not take place in real time: it is usually configured on a daily or weekly basis, but for specific cases even hourly schedule can be considered)

FTP based integrations can also be customized to import external campaign codes (meant as customer's application internal campaign or recipient identities)   without using personal data fields and to export email statistics with their reference to the external campaign code. This feature is not available with MailUp web services. More on this.

REST API

Rest API allows immediate and bidirectional import and export of recipients. Due to restriction on HTTP channel, you cannot pass more than 7MB as a payload of a single call. To overcome this limit, you may either split your list into multiple calls or provide a large file to be uploaded using FTP (the latter is the reccomended option in case of very large data set). The REST API is the only MailUp solution that allows removal of a single recipient from a group.

Action	Comments	REST Web Services
Confirm successful communication with the API	Returns authentication information.	GET /Console/Authentication/Info
Retrieve Lists	Retrieves all lists related information	GET /Console/User/Lists
Retrieve Groups	Retrieves groups for a certain list	GET /Console/List/{ID_LIST}/Groups
Retrieve Fields	Retrieves the actual personal data field names along with ids. Useful for adding or updating subscribers' personal information afterwards.	GET /Console/Recipient/DynamicFields
Bulk Subscribe	Subscribes a set of recipients to a list	POST /Console/List/{id_List}/Recipients or POST /Console/Group/{ID_GROUP}/Recipients
Bulk Unsubscribe	Unsubscribes a set of recipients from a list	POST /Console/List/{id_List}/Recipients?importType='asOptout'
Bulk Update	Updates personal data fields of a set of recipients	POST /Console/List/{id_List}/Recipients or POST /Console/Group/{ID_GROUP}/Recipients
Retrieve list of subscribers	Retrieves recipients that subscribed to a certain list	GET /Console/List/{ID_LIST}/Recipients/Subscribed

Use confirmed opt-in (COI)

Management of new subscribers (single or bulk) is specified in the section above. In some cases you may need to verify if the email addresses and allow subscription only to existing email addresses that belong to subscribers that really want to receive messages from the MailUp list they subscribed to. This case is also called double optin and subscribers have to click on a specific link that is contained in an email message that is sent to them after their signup.

Double opt-in is useful when your application does not check it the email addresses are valid.

HTTP GET/POST and subscription forms

Two HTTP GET/POST methods, (subscribe.aspx and xmlsubscribe.aspx) use confirmed optin by default, they can be used for single recipients. After signup, the subscriber receives the default confimation request message that is configured for the specified MailUp list. Please refer to the documentation of HTTP GET/POST for more details.

REST API

At the moment Rest API supports the COI process by means of a querystring parameter called "ConfirmEmail" that applies to POST calls to Console/List/{ListID}/Recipient and Console/Group/{GroupID}/Recipient endpoints. When this parameter is set as true, the actual behavior can be different if specified recipient already exists:

- new recipient -> added as pending, confirmation request email is sent
- already subscribed -> no change
- already pending -> remains pending, confirmation request email is sent
- already unsubscribed -> moved to pending status, confirmation request email is sent

Manage suppression lists (i.e. "do not email" lists of recipients)

Suppression lists are not natively supported by MailUp but you can create a dedicated group of recipients to exclude from sending. In order to automate the process, you can contact MailUp support department and request for the configuration of an FTP based integration with a batch process runing hourly. This process has to be set to import in a new group (where the destination group name mathces the actual import file name) the recipients that are listed in the file to be imported. Then, when you access to MailUp admin console, you can schedule or fire your campaign to be sent to "all subscribers except for the recipients that belong to a certain group". It should be easy for you to identify the right group because it has the name of the imported file plus a timestamp.

Send transactional and other personalized emails

SMTP or API: it's your choice

Sending messages through MailUp from an external application allows you to take advantage of:

- The high reputation and scalability of the MailUp sending infrastructure
- Automatic bounce and feedback loop management
- Ability to include dynamic tags and dynamic content
- Ability to check the send status via API

You can send transactional emails:

- Using our SMTP relay
- Using the REST API

Message personalization

You can personalize messages using either data that already exists in the MailUp recipient database (and merge tags that will retrieve that data) or by passing value pairs to the system. This is supported using both SMTP relay and the API.

Bulk email messages

This section describes the different ways through which MailUp can be used to send email messages to groups and lists of users without accessing the administration console. The methods described above can be used to send individual, personalized messages, such as transactional emails. However, if the goal is to send large amounts of messages or DEMs, the best practice to follow is described below.

Some possible scenarios are presented here - with the indication of the web services to be used - as examples of common scenarios. This is in no way an exhaustive overview of all the possible scenarios.

Please note that these cases do not talk about authentication and authorization, you should refer to the specific documentation of each technical solutions for these details.

You may want to send a message to a group of users and both the message and the recipients list are already available in the MailUp console.

REST API

1. If the message has to be sent to all the recipients in a list you can call POST /Console/List/{id_List}/Email/{id_Message}/Send.
2. If the message has to be sent to a group of a MailUp list you can call POST /Console/Group/{id_Group}/Email/{id_Message}/Send

Message, group and list IDs can be found at the page Settings > Codes or they can be retrieved using API.
Similar to Case 1, but in this case a group of recipients has to be imported as well. Some of these recipients might be already subscribed in MailUp due to a previous signup to a newsletter, some others might be new subscribers. Here below some alternative solutions are listed, refer to Comparison Matrix to see how many recipients can be loaded with each solution.

REST API

REST API supports only sending to a single group or to the whole list

1. To import recipients into a group of a MailUp list you may use POST/Console/Group/{ID_GROUP}/Recipients, while POST /Console/List/ {id_List}/Recipients can be used to simply import into a MailUp list without specifying a group. Use POST /Console/List/{ID_LIST}/Group if you also need to create a new group.
    a. the list of recipients can be provided in the request body or by specifying the name of a file you have previously uploaded via FTP. Refer to this page for more details
2. Use the methods listed in Case #1 for sending email message with REST API

FTP

You can use FTP-ZIP to upload a zip archive containing the list of recipients, the message content and a file descriptor with the mailing instructions. After the upload, MailUp does all the rest, preparing and scheduling the mailing process. Since the whole process is fully asynchronous, you can add an email address in the file descriptor to be notified when the process is completed

LIST+

List+ allows you to enclose a list of recipients to the message to be sent (by using a file named **one2many.import**) instead of sending to lists or groups that are already present in MailUp. Click here for more information about this feature
In some cases it may be necessary to import into MailUp a message created using an external system.

REST API

With the REST API you can fully handle message tags, images and attachments.

1. To create a message use POST /Console/List/{id_List}/Email, passing message content as parameter. Check out this page to know more about the available options
2. Use the methods listed in Case #1 for sending email message with REST API

LIST+

With List+ you can create an email message with your email client or your application and send it to a MailUp specific email address. Depending on the specified address, this action can lead to

- save email message to MailUp and send it to all the recipients of a MailUp list
- save email message to MailUp and send it to all the recipients of one or more groups in a MailUp list
- save email message to MailUp without sending it
- save email message to MailUp and send it to a recipient for test purposes

This solution is very powerful if you want to work outside the MailUp console account and it has some interesting additional features like automatic upload of images, support for dynamic fields and sender verification. In addition, List+ allows you to enclose a list of recipients (by using a file named one2many.import) instead of sending to existing lists or groups.
Most customers need to track campaign results. When the same message is sent more than once (e.g. three different times, to three different groups), the results are linked to the message ID, not to the individual mailings. If you need to have distinct results for each mailing, you can make a copy of the message before sending it again.

REST API

You can use GET /Console/List/{id_List}/Email/{id_Message} to get message content and then you need to create an identical message with POST /Console/List/{id_List}/Email
If you are in need to send a high number of personalized messages through the API, please refer to the section of "Transactional emails" or contact us.

REST API and List+

You can create personalized sections within your email by means of Dynamic Content feature and MailUp filters. Dynamic Content is not available with email messages sent with HTTP GET/POST API and SMTP+

Send Transactional or Bulk SMS

As for the email channel, the term "transactional SMS" is usually used to distinguish messages that do not belong to "Promotional" or "Bulk" categories.

Please check this page for regulations and limitation when sending SMS globally.

REST API, FTP

The REST APIs offer reliable and scalable solutions for transactional and Bulk SMS. In addition, bulk sending can also be performed with the "SMS sending from CSV file" feature, which uses FTP data tranfer to allow massive sending of highly personalized text messages.

The following table includes the recommended API methods to be used to perform certain tasks.

Num.	Action	Description	REST API	FTP
1	Create new text message	Create a new text message with dynamic fields (optional) that can then be used as "existing message" fro the actions listed below (MESSAGE_ID is returned). New message is also available on MailUp console.	Createamessage	
2	Single sending of EXISTING messages to a specified recipient	A single existing text message is sent to a single recipient by specifying MESSAGE_ID and destination number. Message shall be previously created and it can be customized by means of dynamic fields	SingleSend	
3	TRANSACTIONAL single sending to a specified recipient	High personalization of messages, suitable for high volumes, effective data aggregation, no need to save content on MailUp console	Transactional SMS using APIs	
4	Bulk sending of EXISTING messages to all subscribers or groups of them	A single existing text message is sent to many recipients (e.g. all list subscribers, one or more groups ...). by specifying MESSAGE_ID and List/group identifiers Message shall be previously created and it can be customized by means of dynamic fields	Sendtogroup/list	
5	Bulk sending of NEW messages to all subscribers or groups of them	A new text message (with or without dynamic fields) has to be created and sent to a list of subscribers. Each action of this type creates a new message on MailUp admin console. The greater is the number of messages and the higher is the load on MailUp admin console, leading to slowering access time to web pages and response time of web services. Please use alternative solutions provided in this page if you have to make more than ten actions of this type per day	Createamessage + Sendtogroup/list	
6	Bulk sending of NEW messages to a list of recipients with a specific text message for each recipient	In some cases customization through dynamic fields could be not enough A list of {*recipient + message text for this recipient*} has to be provided as input and all messages shall be sent as a single campaign. This option has not the side effects of option #5 and it also allows sending of completely different messages for each recipient.	N/A	SMS sending from CSV file
7	Immediate sending	For text messages to be sent as soon as possible	Createamessage + Sendtogroup/list	SMS sending from CSV file
8	Scheduled sending	For text messages to be sent at specified time	Createamessage + Sendtogroup/list	SMS sending from CSV file
9	Sending of concatenated messages	When message length is higher than 160 characters (or higher than 70 chars if "Unicode" (use Unicode for non western alphabets)	Createamessage + Sendtogroup/list	SMS sending from CSV file
10	Read Available credits	Retrieve value of remaining credits (either as a whole at account level or by list)	N/A	
11	SMS Price	To know how many credits are used when sending a SMS in any country, visit SMS Credits & SMS Price		

Extract or display statistics

Email statistics are mainly the number and the details about opens and clicks, usually grouped by message. About SMS we can only know the delivery status and, in case of error, the error code.

REST API

REST API offer the best suite about MailUp statistics, also allowing filtered and paged searches. More on this page

FTP

A batch export can be configured to periodically export to a CSV file the statistics of email and/or SMS messages. Export frequency can be set for hourly to forth weekly schedules. Contact us to request for a batch export for your console account (non standard configurations may require an extra fee). More on this page

Frontend

Frontend pages provide a reduced version of some statistics pages of the console account, so that an external application can embed them. Available pages are similar to Summary Report and Opens report.