

Email messages

This page provides you with information about how to use MailUp REST API to create email messages and to send them to specified recipients.

Reference Links

- [Base URL](#)
- [Automated help page](#)
- [All methods](#)

On this page:

- [Email messages](#)
 - [Create a message](#)
 - [Enable Dynamic fields](#)
 - [List Messages](#)
 - [Read message details](#)
 - [Sending Queue Status](#)
 - [Update message](#)
 - [Manage tags](#)
 - [Manage email attachments](#)
 - [Copy message](#)
 - [Manage message online visibility](#)
- [Send Email Messages](#)
 - [Send to all subscribers](#)
 - [Send to list recipients filtering by groups](#)
 - [Send to a group](#)
 - [Send to a recipient](#)
 - [Retrieve sending history](#)
 - [Stop a sending](#)
 - [Delete a sending](#)

Email messages

An email message is always created inside the environment of a MailUp list, so it cannot be managed outside of that list.

Please note that, when creating a new email message, any referenced tag or attachment must be created in advance for that list.

Create a message



Important about creating a new message

1. any referenced tag or attachment must be created in advance for that list
2. to enable the use of dynamic fields, which are disabled by default, choose one of these options:
 - a. specify at least one default value in the "Fields" parameter (as in the sample request below);
 - b. use the [EnableDynamicFields](#) method;
 - c. use the request property *UseDynamicField: true*

Description	Create a new email message
HTTP Method	POST

<p>URL</p>	<p>Create a message</p> <pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Email</pre>
<p>JSON request (example)</p>	<pre>{ "Subject": "Monthly Newsletter 4", "Content": "Hello world", "Embed": false, "Fields": [{ "Description": "Country", "Id": 8, "Value": "USA" }], "IsConfirmation": false, "Notes": "Issue n. 5", "Tags": null, "TrackingInfo": { "CustomParams": "", "Enabled": true, "Protocols": ["http:", "https:", "ftp:", "news:"] } }</pre> <pre>{ "Subject": "Monthly Newsletter 4", "IsConfirmation": false, "Content": "nothing here", "Notes": "notes", "UseDynamicField": true, "Embed": false, "PreHeader": "", "TrackingInfo": { "CustomParams": "", "Enabled": false, "Protocols": [] } }</pre> <pre>{ "Subject": "Monthly Newsletter 4", "IsConfirmation": false, "Content": "nothing here", "Notes": "notes", "UseDynamicField": false, "Embed": false, "PreHeader": "", "TrackingInfo": { "CustomParams": "", "Enabled": false, "Protocols": [] } }</pre>

JSON response (example)	<pre>{ "CreationDate": "2016-09-16 15:57:35Z", "Notes": "", "Subject": "Monthly Newsletter 4", "idList": 2, "idMessage": 1060 }</pre>
Paging and filtering (example)	

Enable Dynamic fields

Description	Enable or disable dynamic fields
HTTP Method	PUT
URL	<p>Create a message</p> <pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/{id_List}/Email/{id_Message}/EnableDynamicFields/{enable}</pre>
JSON request (example)	<p>Enable or Disable</p> <pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/{id_List}/Email/{id_Message}/EnableDynamicFields/true https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/{id_List}/Email/{id_Message}/EnableDynamicFields/false</pre>
JSON response (example)	Message details are returned
Paging and filtering (example)	

List Messages

Description	Retrieve email messages of a specified list
HTTP Method	GET

<p>URL</p>	<p>Retrieve all email messages of that list</p> <pre>GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Emails</pre> <p>Retrieve only email messages that have been published (i.e. with public visibility, see details)¹</p> <pre>GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Online/Emails</pre> <p>Retrieve only email messages that have been archived (see details)¹</p> <pre>GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Archived/Emails</pre> <p>Note:</p> <p>¹ an empty list is returned if public visibility of messages (aka Web library) is disabled. You shall enable it to Retrieve them</p>
<p>JSON request (example)</p>	<p>none</p>
<p>JSON response (example)</p>	<p>All email messages of a specified list</p> <pre>{ "IsPaginated":false, "Items":[{ "CreationDate": "2017-05-29 08:21:59Z", "Notes": "Issue n. 5", "Subject": "Monthly Newsletter 4", "idList": 1, "idMessage": 38, "ServiceType": 0 }, { "CreationDate": "2017-05-24 13:05:08Z", "Notes": "notes", "Subject": "La prova", "idList": 1, "idMessage": 37, "ServiceType": 0 }], "PageNumber":0, "PageSize":20, "Skipped":0, "TotalElementsCount":2 }</pre>

ServiceType field shows the message types. Here are the most common values:

Value	Description
0	Newsletter
1	SMTP+

P.S.: the ServiceType field has a value greater than 1 when your platform has a custom configuration. Please contact us, if you need more information.

List published messages (public URLs are also provided)

"LastSendDate":"0001-01-01 00:00:00Z" is returned when message has never been sent

```
{
  "IsPaginated":false,
  "Items":[
    {
      "Subject":"Monthly Newsletter 4",
      "idList":2,
      "idMessage":62,
      "CreationDate":"2014-05-21 12:14:41Z",
      "LastSendDate":"0001-01-01 00:00:00Z",
      "Notes":"Issue n. 5",
      "Url":"http://\d0i4e.s57.it/frontend\
/nl_preview_window.aspx?idNL=62"
    }
  ],
  "PageNumber":0,
  "PageSize":20,
  "Skipped":0,
  "TotalElementsCount":1
}
```

List archived messages

```

{
  "IsPaginated":false,
  "Items":[
    {
      "Subject":"Annual Newsletter n.13",
      "idList":2,
      "idMessage":27,
      "CreationDate":"2014-05-22 09:37:56Z",
      "LastSendDate":"",
      "Notes":"Issue n. 10 June 2014",
      "Url":"http://\d0i4f.s57.it/frontend\
/nl_preview_window.aspx?idNL=27"
    },
    {
      "Subject":"Annual Newsletter n.4",
      "idList":2,
      "idMessage":26,
      "CreationDate":"2014-05-22 08:55:15Z",
      "LastSendDate":"",
      "Notes":"Issue n. 6 June 2014",
      "Url":"http://\d0i4f.s57.it/frontend\
/nl_preview_window.aspx?idNL=26"
    }
  ],
  "PageNumber":0,
  "PageSize":20,
  "Skipped":0,
  "TotalElementsCount":2
}

```

Paging and filtering (example)

2 items per page, get first page (count starts from zero):

- <https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?PageNumber=0&PageSize=2>

Retrieve messages that contain the word "test" in the subject field and order by ID. Note: search by text is case sensitive:

- [https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?filterby="Subject.Contains\(%27test%27\)"&orderby="idMessage+asc"](https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?filterby=)

Retrieve only the messages with tags whose IDs are 1, 13, or 45 (at least one of them):

- [https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?tags="1,13,45"](https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?tags=)

Retrieve only the messages with tags whose ID is 13 or names are *customers*, *welcome message* or *hello, world* (at least one of them):

- [https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?tags="13,customers,welcome message,'hello, world'"](https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?tags=)

Retrieve only the newsletter messages:

- [https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?filterby="servicetype==0"](https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?filterby=)

Retrieve only the SMTP+ messages:

- [https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?filterby="servicetype==1"](https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?filterby=)

Read message details

Description	Retrieve the email details (content, settings, attachment) by specified id
HTTP Method	GET

URL	<p>Retrieve the email details by specified id</p> <pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Email/{id_Message}</pre> <p>Get attachment list for the specified message-id</p> <pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Email/{id_Message}/Attachment</pre>
JSON request (example)	<p>none</p>
JSON response (example)	<p>Message content</p> <pre>{ "Subject": "Monthly Newsletter", "idList": 2, "idMessage": 24, "Content": " <table><tbody><tr><td>Image&nbsp;</td><td>Text&nbsp;</td></tr><tr><td style="width: 200px; height: 70px;">&nbsp;</td><td>&nbsp;<This screenshot shows you how to perform login with social networks
&nbsp;<(see manual&nbsp;</td></tr></tbody></table> ", "Embed": false, "Fields": [{"Description": "FirstName", "Id": 1, "Value": ""}, {"Description": "LastName", "Id": 2, "Value": ""}, {"Description": "Company", "Id": 3, "Value": ""}, {"Description": "City", "Id": 4, "Value": ""}, {"Description": "Province", "Id": 5, "Value": ""}, {"Description": "ZIP", "Id": 6, "Value": ""}, {"Description": "State", "Id": 7, "Value": ""}, {"Description": "Region", "Id": 8, "Value": ""}, {"Description": "Address", "Id": 9, "Value": ""}, {"Description": "Gender", "Id": 10, "Value": ""}, {"Description": "phone", "Id": 11, "Value": ""}, {"Description": "CustomerID", "Id": 12, "Value": ""}, {"Description": "LatestOrderID", "Id": 13, "Value": ""}, {"Description": "LatestOrderDate", "Id": 14, "Value": ""}, {"Description": "LatestOrderAmount", "Id": 15, "Value": ""}, {"Description": "LatestOrderProductIDs", "Id": 16, "Value": ""}, {"Description": "LatestOrderCategoryIDs", "Id": 17, "Value": ""}, {"Description": "LatestShippedOrderDate", "Id": 18, "Value": ""}, {"Description": "LatestShippedOrderID", "Id": 19, "Value": ""}, {"Description": "LatestAbandonedCartDate", "Id": 20, "Value": ""}, {"Description": "LatestAbandonedCartTotal", "Id": 21, "Value": ""}, {"Description": "LatestAbandonedCartID", "Id": 22, "Value": ""}, {"Description": "TotalOrdered", "Id": 23, "Value": ""}, {"Description": "TotalOrderedLast12m", "Id": 24, "Value": ""}, {"Description": "TotalOrderedLast30d", "Id": 25, "Value": ""}, {"Description": "AllOrderedProductIDs", "Id": 26, "Value": ""}, {"Description": "WantsBlogUpdate", "Id": 27, "Value": ""}], "IsConfirmation": false, "Notes": "Issue n. 5 May 2014", "Tags": [{"Enabled": false, "Id": 5, "Name": "news"}, {"Enabled": false, "Id": 4, "Name": "users-manual"}], "TrackingInfo": {"CustomParams": "", "Enabled": true, "Protocols": ["http:", "https:", "ftp:", "news:"]}, "Attachments": [{"Name": "24\\jp.txt", "Path": "http:\\/\\/d0i4f.s57.it\\d0i4f\\download\\24\\24\\jp.txt", "Slot": 1}]} </pre> <p>Message attachments</p> <pre>[{"Name": "24\\jp.txt", "Path": "http:\\/\\/d0i4f.s57.it\\d0i4f\\download\\24\\24\\jp.txt", "Slot": 1}]</pre>

Paging and filtering (example)	<p>2 items per page, get first page (count starts from zero)</p> <ul style="list-style-type: none"> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?PageNumber=0&PageSize=2 <p>Retrieve messages that contain the word "test" in the subject field and order by ID. Note: search by text is case sensitive</p> <ul style="list-style-type: none"> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/2/Emails?filterby="Subject.Contains(%27test%27)"&orderby="idMessage+asc"
---------------------------------------	---

Sending Queue Status

It is possible to retrieve the status of the sending queue for the outgoing messages by calling the following methods:

- 1) status of ongoing sendings: GET <https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Email/Sendings/Immediate>
- 2) status of waiting sendings: GET <https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Email/Sendings/Deferred>
- 3) status of sending to be planned: GET <https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Email/Sendings/Undefined>

Moreover, it is possible to retrieve the first available date to send a specific sending:

GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Email/Sending/{id_sending}/Deferred

where {id_sending} is the unique identifier of the sending to consider. The result is the date UTC date specified with the format described in the section 'Type and cast definition' of this page: ["Paging and filtering"](#).

Update message

This method allows you to modify an existing message (content and settings).

Please note that

- The "Tags" field is mandatory, if you leave it empty the existing tags of that message are not modified. There is no way to clear all tags
- "Fields" parameter is optional
- "TrackingInfo > Protocols" field values are ignored if "TrackingInfo > Enabled" is set to false

Description	Modify an existing message
HTTP Method	PUT
URL	<code>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Email/{id_Message}</code>
JSON request (example)	<p>Request data format is the same as the response of "GET /Console/List/{id_List}/Email/{id_Message}"</p> <pre>{ "Subject": "Annual Newsletter n.9", "idList": 2, "Content": " <table><tbody><tr><td>Image&nbsp;</td><td>Text&nbsp;</td></tr><tr><td style="width: 200px; height: 70px;">&nbsp;</td><td>&nbsp;This screenshot shows you how to perform login with social networks
&nbsp;(see manual&nbsp;</td></tr></tbody></table>", "Embed": false, "IsConfirmation": false, "Notes": "Issue n. 9 June 2014", "Tags": [], "TrackingInfo": { "CustomParams": "", "Enabled": true, "Protocols": ["http:", "ftp:", "news:"] } }</pre>
JSON response (example)	<pre>{ "Subject": "Monthly Newsletter!", "idList": 2, "idMessage": 24 }</pre>

Paging and filtering (example)	none
---------------------------------------	------

Manage tags

Description	Read available tags for a specified list
HTTP Method	GET, POST, PUT, DELETE
URL	<p>Retrieve list tags:</p> <pre>GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Tags</pre> <p>Create a new tag:</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Tag</pre> <p>Modify a tag:</p> <pre>PUT https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Tag/{id_Tag}</pre> <p>Remove a list tag:</p> <pre>DELETE https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Tag/{id_Tag}</pre>
JSON request (example)	<ul style="list-style-type: none"> No request body is required for GET and DELETE methods POST method requires a simple string as parameter (e.g. "my-new-tag"), not a JSON structure PUT method requires a full structure like {"Enabled":true,"Id":6,"Name":"my-new-tag"}
JSON response (example)	<p>GET method returns the retrieved tags list</p> <pre>{ "IsPaginated":false, "Items":[{ "Enabled":true, "Id":5, "Name":"news" }, { "Enabled":true, "Id":4, "Name":"users-manual" }, { "Enabled":true, "Id":3, "Name":"\$5 off promo" }, { "Enabled":true, "Id":2, "Name":"preferred customer" }, { "Enabled":true, "Id":1, "Name":"promotion" }], "PageNumber":0, "PageSize":20, "Skipped":0, "TotalElementsCount":5 }</pre> <p>POST and PUT methods return the full structure of the created/updated tag, e.g. {"Enabled":true,"Id":6,"Name":"my-new-tag"}</p> <p>An empty response is returned by DELETE method</p>
Paging and filtering (example)	none

Manage email attachments

MailUp messages support a maximum of five attachments. Each attachment is placed in a specified "slot": adding and removing an attachment means adding to a message "slot" or removing from it.

/56VX6gldBXRuY3U9rgWgi9yCLC3Xv1ILaZkzDhmWYwap0LC4Xa0M1OI7Q1rSbevkSLlWdMJzLW9FTuJ6XmGPi0
F3bplNs72A3Vcc3YJiFFOKjEi5GunJdrk3JPYd
/NNv0Tbb2LWYHUvo8ZhkJD2yNcCO0CC5Zw6FwsYo7fch+FXfjnlyHLOZIJauCLpQXlrNg17HDzmj9G9xt2gqy
I5Kt/GeoYOKDKZp9HooB6nPIJ9/njuQS
/VYMclYHTzHTqkhiMsc19MgBbIOG27Shj1ghzqQbgjY5Yy55H0pxbcAsjYy+mJheXua0nc3cfUAGtAAAtvvKWAU1
M1jfrA0NHsAsF6oCIiAiIgiIiICIsbE6sYfhsKruUbHPPuBKcueZx
/6t44CMbtFQyH+WMjpP7XKynIKunASjdi+fZq1+
/Rse8n+OR1h8NSnHO+J+RspVU3WyJlvvEWb8SEFU874p5ZzVtdT5XW+6Dpb8AFMXg3YV0WE1VSRvI9sbT6mgk2
9pd8FAKtBlFoyjwbbIRZzaZ85+84FzR8QEEBcRcS8rZ4q5RyMpaPY2zB8GqU
/Bswu1Pv1J6y2Ee4anf3uUfvcXvJ05JuVZ3htEMrcJGzO2
/2n1Lr+sEt+AggziPivlfP1W8G7WYGjvsZ5v5gqQfBswvVUVdSRyDYmn2+c78mqFzPMP5zjdzIST2km5KsNkL
EGZD40CqLHnPLpA39dznFstfeGg+oXQbbMXDjBo5ZKmsuzW4ve+SdzQXE3PN3wC5mhz7geQyPwYfHLK55BcW3IJ
AOnz5De255A81DuY8w1oAMTmtS8vteZrR6LQTs1jeof/jdStk
/gowYeKje3uHm6zDgbWFr2e7ne3MNtbtQRlnfNk2ccbM8wDdgljG7hjRles3JJP/0u38HXc
/Gs2SzkBQxWH3nmw+UOUX1z2yVshjG1he4tb2NudI37ArBcBKQYRkKeqft0j3vv
/BG239wegjDjTinlTiFPb0YtMI/lHnfmXLp
/Bvwp8eqag8oohGPa91++zD3qJsRqzX4hJK70pHuefa4kn81YzgrQjCeHrp3bdK98pv+q3zR7vNJ96CKu00KeU
eIUwHowtBpCnTvmce5bzwcsl8ZzNPOeUMWkfeef8ADT3qMMYrjieLsZ05ySof3klWE8HzDvEckvmdsZpXouf1W
gNHxDkEd8f8V8ezx0QN2wrtZ/M7znfm05fXg+4X47ngyKXETnX7Hos0fAuXD5txLxmepnfvfjXU3hbnT8LKB
/AAccL6DLs9QRvLLoB/hYP8uPcg47wMT8bzhHCDtDCP6nEk/ANWDwGwvynhljyLthjdJ79mt
/u+C5bPwKewc4VU17h0rtP3QdLfgApf8G7C+hwuqqXD03tjaT2NBLvi4dyDlFCGxxzObIqDoIgd6n8P4
/LpWv4G4MMWz5G5wuyBrpT2ahsz4m/uXL5zxTyImupn6nyuLfuG2Z8oCmHwbMM0YbV1B5ue2IexrdR
/vHcg8PcVxG0VHTjms+U+qwdW993dyi/hxhnljPFJERcGUOPsb55
/tW744r5t4gzAHZyWthHuF3fm4j3LeeDlhfjGZ55yNootI+88j
/1a7vQWGe4MYSeQFyq28Om6p4yun07RJLU+4EiPuLm9ynHiLinkbJFXLexERa37zVmb8XBRh4NeG
/95UEfqrNPe5//ognJERAREQEREBERAXC8ast8m8PKix86XTEP5ndV8oK7pQ74SfSeqQW3/F0rtX3tPmfDUg9
/BywrbxBLM9QRvNLpB/hYNvmc5ZPhD4n4pkxkQO80wB+60Fx+Olbzg1JE
/h1SiIjYOD7dT9RL7+8394X5xD4dszZuQknfGImuAa1o07iLnc+odyCsGC0BxTGIYW85JGs29bgFYjJtXjCeHg
gzT0r2RAD9VvnH3eab71+ZV4NQZdzBFUioKMRlG1zQATpIG4Pve/uW84h8P2Z5dD0k74mxarNY0G5dpuTc/w
/mgqvQJq+ujiZ6Uj2sHtcQB+aslxpQ24FwzMDDBWY4G2
/Vfi75WEe9Y2W+C1PgWOW1HjEkhidrDXNABI5bg9Rsfucj4hZFBniGj53xNjJdZrQ65IAub9g
/NBVCldgVVLWN3c9waPatYKZuPzThWAYbSMFomNPsuxrGt+BPeuiwLghT4TjMM
/jmj+1kbIGlOAJabi5B7QF2GfsmQ51wgrSEse06o5G7lptY3H6QPWEFTMLg/EMTi106ujka
/SevS4G3vtZTBnrjTHi2XHwUkUjHyt0vfvJYaGn0g2xNyRtFZYzeANV09jVQa00NcT/Ta3xxa0
/Baigy4+ApD00mnVUEAushA6WN5Mbt7e01BWLWRzV/0jwOEQ2eYWRbbedIQZPzCvrmcAadJwfHJdf0G
/5XcZ9yW30eFxxQvmdEyn+vzWg6jPif78rXKCPDwl7gBuTsFZ7Nrh1Hg06MboFO2AdR1PAa4+3clafDeBNPRYthi
aqV2h7X6Sx0BsQbHfkbLtc/ZQbnTCWQpmdExsgkOgA3IBABv1b3QVDVqKppyjwdIGz4qS387m2v
/AFOXN0faamp6tjzVSuDXBxawNANiDY79aLLGMmJxjCpIjReORhY4DbY9nYRzQUpU15Y4rS4DkoUUFODKA
/TLq5ai5xd0DdyL9vVdbuv4AzippQVURjvt0jXNcB69IiJ71l1mBcGafDMdmjK51LRmzoZMG+g0kamsaTYXASd7
E8kFbVZPBf8ApDgVrvZ5p3SAnbz5b07U3uWrHACnv8A95L/AEN/yvbwk8YdlCnpWbdI8bfwrt
/yWoK9KRMgcUn5My7LTTtEjnPmjHl+kNJa0G40nUPNvzC4rAsOOL41DA3nLlneRc+4bq8898FmY1X9NRSNhc4D
Wx4OgkbagRctJ6xayCv9bV0rax8jzd73F7j2ucSse8qznBHLbsAyY10gIlqHdK4EWIba0bt7hf+YrR5L4IxYVWt
lrZRO5pBbGwFrLjCfx07/Zs02610CwQRP4ReKeLZUigB3mlur
/CwXPzFq3nBLDPJvD2AkedKXSn+Y+b8oCi7j1iPlnPUNNG406JY7De0j3ec07QrA4PQjDMJhbs2ONrB/KOD
/AMIMxERAREQEREBERAWnzbl2LNOBSU01w1+4cObXDDrh7D3i4WRBWS5Lx7I9c8UfSuY79OnIc1w6i5h3B9o96
9vKoaury/Cb9KnzMeLtwHApqlw1CJhfpva9uTb9VzsuQwfiTJVvYsq1FDJAKXuqf
/SNkBFr7gNBA5d42QR15RzV2Vf4TfpTyjmrsg/wm
/SpJouI9Ti1RN4phck8UUzoekE7Wgltuot22IPXzXS50zOMqZbdUvJlyCxoJdrEucQLA2PLc8upBCPlHNXZV
/hN+1PKOauyr
/Cb9K1XceID5swG1q6J9LJ0JnaTI2QFoFzewFuR71jzB4g1eYWRsRYVJ4vi4Dpj02wbqs5ltNyBv3IIz8o5q7Kv
8Jv0p5RzV2Vf4TfpU2ZJzT/AksopZwXdGxkzomnVqlhtvPGwte
/Ldc5TcRqrE6iYUmFyVEcUroukbM1ocWnnZzdrix6+acNvKWauyr/AAm/SnlLNxZV
/hN+1SxS8Q4300IyyRGntC8xnx29XSEXat5o03Nh181gZg4o+QsCpZn0jJNURMUwiTeOPaz300ddxtYcz2II28pZ
q7Kv8Jv0p5SszV2Vf4TfpUs534hR5S8U1RGQ1J6n6dDFNu70Tf0vUvSuzrI3HaqmpqR1Q+mjjedMgbqc8ts3dthZ
p1Xv1HZBEX1LNxZV/hN+1Y9fj+ZsOpHSSuqWrtF30dG0Ae06VLOTM
/1GagsacOeyDW6N8xma4Mc0XiTpDlDbtWr8IXFPE81tiB3n1AP3WguPxD9BquBma8QzJfQKmyYwsjB85rQQ8
uGmxaB1Blx7F2XEfidDKuhts+qcP8Abi/9n25N+J6vVEOSM5x5CyJI6PTJW1MhLG9TGNGlRn+
/UQ3r9m6xsh5GquIuLuqat7+gLrySu9KQ9bGX29Vxs3q7EH7hub8xZg1yUz53tlb9FG3SCf0R5qzfkGauyr
/Cb9Kn
/DcPgwLDGxQtbfCWAGwHaSesnmSeayKarZVR1zHBzQbXBUFHFN9t+aeG7bq8eUc1dlX+E36VpMwYj2Y5G0q4K
mUSBddTALA8+QHYrRQ1TKhxDhtCzR0kG3cvZJZecLLOqpmF5RxcjCMQZNBsVDJWG7XC09jYjrBHIrpfKWauyr
/AAm/SrHiPqrj5SzV2Vf4TfpXy+szTUN0nxvfbzjw/ENF9WQQmWQvwt4Ty0GKnrMR/5GnVHFqDzq
/XkIuCQdwAee57FNCxqiuJp2ju7d3ogbl3sA3KxqnFD0miFhkk6xezWfdlH1c1nlrYY9a0x0s8ukbJF8xk1gluL
G24G+
/t619LRmIiICiAiIgiIiI646Vwp8mCHUGmomZHC7ANvqcT6rBcdhFXHhmcZ5aeodW09Lh7j08tn9E4A6WROADR
fYWA7exTdv0Mva0CWnJwOwtodb2XGy
/GyDdHTGMRRiM82BoDT7RaxQQ3wgpaxXwLAXSUVD3GR1IyQBrnbkhdzNz5oud+pdBxga7F8WwygY4tM9R0jnCxL
WSA3seZs4kfdUgU+E09LMHRwxMcOTmsa0jt3AXu+1ZJUNeWNL27NcQCR22PMIITq6k4YmfklkM8sbGUrKmt07Se
aWAdzWgc9gOS2WXZqXAcqPkhxSWaSGje80wkBja5zdOzQ0HZ7wBc8yFL78LgkDrwxnWdTrsBlHqJ230

```

/Mr5Zg1MxpAgiAcLOAJaLi4NjtvuAfcgjbh9mWiwLhkGRzXuqI6eSd0bTd2qznEEdouAtTwhpqY0tL
/8AKSipe90rqRkgDS67jZzdNzdoBO
/apeZgtNHe0EiuLG0bRcdh23C+6fCYKWUOjhiY4ciljWnvAQQLSReWs7VlAdoDXPqql3V0Ud
/NPqLj+Sx8yV9TjmFYjX+Kf1LOGwQzF7W9FDHKAAGekdTgLu+u6sI3DoWveRFHd9w86R5wPPVt53vX0aGI0nRdGz
o+WjSNPb6NrIIQzRH/qjEJ+Tm0mDtd7Jhta8fKtrw
/xVzcmYrisg0vlc63sZGA238ziPcpajoIotWmNg1ANdZoFwBYA7bi211+Nw+JtJ0Qjz0Z
/Q0jTzv6Nrc0HLcIMO8m8PaUdcjTmf5zqHwIUZeEnK92NUjSD0YicWnqLi4ax7gG94U+xsEbAGgAAWAGwA6gF51
VJHWR2kY14vez2hvw27oK5cKuFjSyubU1YLKtMlu7XS+zraz+LmertVjaSmZR0zy42hjGgNalosABYAC9WtDW2G
wC/UHKY/TOMBnzHW5ztEEQPmgk2a4/rHrX3h9F4y/xce9BDtJY26SQ7kXG9gsimd5VzE5
/OOAAw+t55n3f4Xz1+ujocHc6Rwa7W8vB56tR6uZ6lx5hhdXiv8bvzveY9ef8Ad9p9HUuWc0+GdZty+lvT0nvfq
+vFmUuaYhE0NHRO1Bott1X962M1cfKjYWNBNtTyT6Ler2krFwOJ1RUPqZAWmSwY082sHK/rPNYFc+ajZUODdJe
//kJBLhsI2sHb6zy3W0zump8cm0tt6dp+9vdlcJqZ8Nu9k28/83batxhkF04sIe406MNH6
/Zf1cyvzBq91TQufLpADiA5uzS0fpc
/Ve4v61rZMDMNPABH0oYHam6g273W84k8xzHcsqpWz+JUxEz901msj9FvYT1vPcPurTPxFz3s6Tp29fb9IuGhMd
pe/Xv6fPN4T40+WqYWHTG5+lg0hzpd
/OdYkaWdtXtjeNiKncyIGR58y7eQJ2AJ6z6gsiHaohThrxr013HY7CwAtybbawWYKGMBlmgBhu0DYA2I5e9TNLx
Fxu+W2/t+PnmrdTQmU2m+3v8An55NXhuBijw8h7z0hZpMgO7R2NvyAWPh/wDtVDI6Ww0jQ7/cJa0sA6
/ODQS4+0+tbysomVrAJBqaDe19j7R1r2jjETAGgAdkBsrzws1kx5Sfff56q3xNstY52
/bb56PpERex5RERAREQEREBERAREQEREBERAREQEREBERAREQF51DS6BwaboINj2G2y9EUWbxMuzW5ew84bhjWO
9O5c62+5Pblr3FhsM1TrMbC/9YtF1los8dHDHCYbcovlq53K5b86LVVf
/wDbjMcf6MQ6V3t5Rj8z7ltV8NjDZC4Dc2ue23JTqYcck7d0YZ8NtfaIi0UEREBERAREQEREBERAREQEREBERAR
EQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBER
RAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBERAREQEREBER
/9k=", "Name": "10-years.png", "Slot": 2, "idList": 2,
"idMessage": 24}

```

JSON response (example)

The GET method returns a list of the documents that are attached to the specified message. An empty array ([]) is returned if no document is attached.

In this case, the specified message (with ID=28) has five attachments

```

[{"Name": "28\\readme.txt", "Path": "http:\\\\d0i4f.s57.it\\d0i4f\\download\\28\\28\\
/readme.txt", "Slot": 1}, {"Name": "28\\optin1.csv", "Path": "http:\\\\d0i4f.s57.
it\\d0i4f\\download\\28\\28\\optin1.csv", "Slot": 2}, {"Name": "28\\product-logo.png", "
Path": "http:\\\\d0i4f.s57.it\\d0i4f\\download\\28\\28\\product-logo.png", "Slot": 3},
{"Name": "28\\Feature User Guide 1 0.pdf", "Path": "http:\\\\d0i4f.s57.
it\\d0i4f\\download\\28\\28\\Feature User Guide 1 0.pdf", "Slot": 4}, {"Name": "28\\max.
jpg", "Path": "http:\\\\d0i4f.s57.it\\d0i4f\\download\\28\\28\\max.jpg", "Slot": 5}]

```

POST method returns the list of attachments after the command execution

```

[{"Name": "24\\jp.txt", "Path": "http:\\\\d0i4f.s57.it\\d0i4f\\download\\24\\24\\jp.txt", "
Slot": 1}, {"Name": "24\\10-years.png", "Path": "http:\\\\d0i4f.s57.
it\\d0i4f\\download\\24\\24\\10-years.png", "Slot": 2}]

```

DELETE method returns details about remaining attachments after deletion

The same result is returned even if an empty slot is specified (of course in this case no deletion occurs)

```

[{"Name": "24\\10-years.png", "Path": "http:\\\\d0i4f.s57.it\\d0i4f\\download\\24\\24\\10-
years.png", "Slot": 2}]

```

Paging and filtering (example)

none

Copy message

Description	Copy an existing message
HTTP Method	POST
URL	<p>Create a message</p> <pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/{id_List}/Email/{id_Message}</pre>
JSON request (example)	<p>Copy message as-is:</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/2/Email/135</pre> <p>Copy message modifying Notes:</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/2/Email/135</pre> <pre>{"Notes": "This is a new note!!!"}</pre> <p>Copy message modifying Subject:</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/2/Email/135</pre> <pre>{"Subject": "This is the new subject!!!"}</pre> <p>Copy message modifying Subject and Notes:</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/2/Email/135</pre> <pre>{ "Subject": "This is the new subject!!!", "Notes": "This is a new note!!!"}</pre>
JSON response (example)	<pre>{ "CreationDate": "2017-08-30 15:24:07Z", "Notes": "This is a new note!!!", "Subject": "Important email", "idList": 1, "idMessage": 136}</pre>

Paging and filtering (example)	
---------------------------------------	--

Manage message online visibility


Description	Publish an existing message
HTTP Method	PUT
URL	<p>Create a message</p> <pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/{id_List}/Email/{id_Message}/OnlineVisibility</pre>
JSON request (example)	<p>Publish a message:</p> <pre>PUT https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/2/Email/135/OnlineVisibility true</pre> <p>Unpublish a message:</p> <pre>PUT https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/list/2/Email/135/OnlineVisibility false</pre>
JSON response (example)	None
Paging and filtering (example)	

Send Email Messages

This section contains methods to send messages and to retrieve the history of sent messages.

You can enrich the calls to send messages using some query string parameters, as follow:

Querystring parameter	Description	Example
SenderName SenderAddress	Email messages are usually sent with the sender name and the sender name that is specified as default settings for the MailUp list. As an alternative, you can specify a different sender for a certain sending by specifying "SenderName" and "SenderAddress" parameters. You have to provide <i>both parameters and not only one of them</i> .	POST ... /Send? SenderName= 'John Smith'&SenderAddress='smith@example.com'
DateTime	It is possible to specify an eventual UTC date/time for a deferred sending. When a message send is scheduled you cannot modify it. The format described in the section 'Type and cast definition' of this page: " Paging and filtering ".	POST ... /Send? datetime=" 2014-10-20 05:00:00Z"

	<p>It is possible to specify the value undefined for the DateTime query string parameter. It means the message sending is scheduled as inactive. This option allows you to create a relationship between the message and the recipients, as a snapshot: you can modify the message and when the message is ready you can decide to schedule the newsletter or to send it immediately.</p>	<p>POST ... /Send? datetime="undefined"</p>
	<p>It is possible to omit the DateTime query string parameter (default) or set its value to now. It means the message is enqueued and it will be sent when the dequeue process reaches it.</p> <div style="border: 1px solid #f0e68c; padding: 10px; margin: 10px 0;"> <p> Important tip!</p> <p>As you can imagine, if you use this option to send to some recipients and you use the method to send to a single recipient now, you append a lot of items to your sending queue. Every time an item is processed, it starts a verification process that requires some seconds: if you enqueue 10.000 items your sending period could be very long. So, if you need to send to a single recipient we suggest using our Transactional Emails APIs; otherwise, please use a list or groups to enqueue multi-recipients items to the sending queue.</p> </div>	<p>POST .../Send POST ... /Send? datetime="now"</p>
replyto	<p>It is possible to specify an email address for a reply.</p>	<p>POST ... /Send? replyto="replyto@yourdomain.it"</p>

 Message sending fails with HTTP 500 error if no authorized sender email nor sender email are provided. Ensure that sender name and sender email have been configured at list level (starting from MailUp version 8.6.1 the sender address has also to be a verified address)

Send to all subscribers

Description	Send a message to all list subscribers
HTTP Method	POST
URL	<p>Use the default list sender</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Email/{id_Message}/Send</pre> <p>Use the sender that is provided by query string parameters</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Email/{id_Message}/Send?SenderName="{name}"&SenderAddress="{email}"</pre>
JSON request (example)	none
JSON response (example)	<p>See notes at the bottom of this page¹</p> <pre>{"Id":2147483647,"InvalidRecipients":[],"Sent":7,"UnprocessedRecipients":[],"idMessage":24}</pre>
Paging and filtering (example)	none

Send to list recipients filtering by groups

Description	<p>You may need to restrict your mailing to those recipients that belong to some MailUp groups or you may need to exclude the recipients of some groups.</p> <p>When using specific query string parameters with a valid value, the Send method allows including or excluding the recipients that belong to a list of specified groups.</p> <p>Here is how it works:</p> <ul style="list-style-type: none"> ▪ use the inGroups query string parameter to set the IDs of the MailUp groups to be included (i.e. all the group members will be added) ▪ use the notInGroups query string parameter to set the IDs of the MailUp groups to be excluded; <p>Please note that all the parameters' values must correspond to groups that belong to the MailUp list {id_List} specified in the REST endpoint.</p>
HTTP Method	POST
URL	<p>Inclusion</p> <p>Include recipients that belong to one or more groups. It is mandatory that at least one of provided groups belongs to the specified MailUp list, otherwise the method returns a 500 error. In case of empty values, the inGroups parameter is ignored and the message is sent to all the MailUp list recipients. If a recipient belongs to more than one group, it will be added just one time.</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List} /Email/{id_Message}/Send?inGroups=12 POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List} /Email/{id_Message}/Send?inGroups=12,23,45</pre> <p>Exclusion</p> <p>Exclude subscribers that belong to one or more groups. If the specified MailUp list does not contain all the provided groups, the method does not exclude any recipient and sends the newsletter to all the MailUp list recipients.</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List} /Email/{id_Message}/Send?notInGroups=12 POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List} /Email/{id_Message}/Send?notInGroups=12,23,45</pre> <p>Inclusion and exclusion</p> <p>This is a case that merges the cases seen above: both <i>inGroups</i> and <i>notInGroups</i> are used.</p> <p><i>inGroups</i>: at least one of the provided groups must belong to the specified MailUp list. Otherwise, the method returns a 500 error. In case of empty values, the inGroups parameter is ignored and the message is sent to all the MailUp list recipients.</p> <p><i>notInGroups</i>: if the specified MailUp list does not contain all the provided groups, the method does not exclude any recipient and sends the newsletter to all the recipients filtered by the <i>inGroups</i> parameter.</p> <pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List} /Email/{id_Message}/Send?inGroups=12&notInGroups=89,90 POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List} /Email/{id_Message}/Send?inGroups=12,23,45&notInGroups=89</pre>
JSON request (example)	none

JSON response (example)	See notes at the bottom of this page ¹ <pre>{ "Id":2147483647, "InvalidRecipients":[], "Sent":7, "UnprocessedRecipients":[], "idMessage":24 }</pre>
Paging and filtering (example)	none

Send to a group

Use the default list sender or a sender that is provided by query string parameters (see example above)

Description	Send a message to all subscribers of a specified group
HTTP Method	POST
URL	<pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Group/{id_Group}/Email/{id_Message}/Send</pre>
JSON request (example)	none
JSON response (example)	See notes at the bottom of this page ¹ <pre>{ "Id":2147483647, "InvalidRecipients":[], "Sent":3, "UnprocessedRecipients":[], "idMessage":30 }</pre>
Paging and filtering (example)	none

Send to a recipient

Use the default list sender or a sender that is provided by query string parameters (see example above)

Description	Send a message to a single recipient
HTTP Method	POST
URL	<pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Email/Send</pre>
JSON request (example)	<pre>{ "Email": "cheetah@jungle.net", "idMessage": 30 }</pre>
JSON response (example)	See notes at the bottom of this page ¹ <pre>{ "Id":2147483647, "InvalidRecipients":[], "Sent":1, "UnprocessedRecipients":[], "idMessage":30 }</pre>

Paging and filtering (example)	none
---------------------------------------	------

Retrieve sending history

An email message can be sent several times. This feature allows retrieving details about each sending.

Description	Retrieve information about the sent history
HTTP Method	GET
URL	<pre>GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/List/{id_List}/Email/{id_Message}/SendHistory</pre>
JSON request (example)	none
JSON response (example)	<p>An email message has been sent three times with different senders, here below the results</p> <pre>{ "IsPaginated":false, "Items":[{ "Id": 26, "EndDate":"2014-04-01 13:53:06Z", "Kb_transferred":6302, "Recipients":1, "SenderEmail":"marketing@example.com", "SenderName":"Your Dealer", "StartDate":"2014-04-01 13:52:35Z", "idMessage":12 }, { "Id": 27, "EndDate":"2014-04-01 13:54:32Z", "Kb_transferred":12604, "Recipients":2, "SenderEmail":"brown@example.com", "SenderName":"Mike Brown", "StartDate":"2014-04-01 13:53:57Z", "idMessage":12 }, { "Id": 28, "EndDate":"2014-04-06 13:48:33Z", "Kb_transferred":12604, "Recipients":2, "SenderEmail":"sales@example.com", "SenderName":"Sales", "StartDate":"2014-04-06 13:48:04Z", "idMessage":12 }], "PageNumber":0, "PageSize":20, "Skipped":0, "TotalElementsCount":3 }</pre>
Paging and filtering (example)	N/A

1. **"Id"** is the sending identifier. **"Sent"** is the size of the sending queue and may include either previously enqueued recipients or, despite its name, discarded recipients (invalid or unprocessed). **"InvalidRecipients"** is expected to be always empty when sending to all list subscribers or a group. **"UnprocessedRecipients"** contains a list of recipients that are excluded from sending due to a restriction (e.g. in case of a trial console account or when the rate-limiting described in API use policy is exceeded).

when creating a new email message, any referenced tag or attachment must be created in advance for that list.

Stop a sending

Use this endpoint to stop a specified sending. It works on scheduled or immediate sendings. It does not change the status of undefined / standby sendings.

You can perform a max 1 request every 30 seconds.

Description	Stop a sending
HTTP Method	POST
URL	<pre>POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Email/Sendings/{id_Sending}/Stop</pre>
JSON request (example)	none
JSON response (example)	none
Paging and filtering (example)	none

Delete a sending

If you've already stopped the sending, use this endpoint to delete it. You **MUST** call the [Stop a sending](#) endpoint before deleting *scheduled* or *immediate / in-progress* sendings. A standby sending does not require to be stopped.

You can perform 1 request every 30 seconds max.

Description	Delete a sending
HTTP Method	DELETE
URL	<pre>DELETE https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Email/Sendings/{id_Sending}</pre>
JSON request (example)	none
JSON response (example)	<pre>200 OK { "IsDeleted": false } 200 OK { "IsDeleted": true }</pre>
Paging and filtering (example)	none