

Text messages - SMS

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

On this page:

- [Subscribe on SMS channel](#)
 - [Add recipients](#)
 - [Read recipients](#)
- [Text messages](#)
 - [Create a message](#)
 - [List Messages](#)
 - [Read message details](#)
 - [Update message](#)
 - [Delete Message](#)
- [Send a text message](#)
 - [Single Send](#)
 - [Send to group/list](#)
 - [Manage Send Queue](#)
- [SMS Statistics](#)
 - [Summary report](#)
 - [Detailed report](#)
- [SMS Credits](#)
 - [Summary report](#)

Subscribe on SMS channel

Add recipients

In the case of already existing recipients the following behavior applies to subscriptions:

- already subscribed remain subscribed
- pending or unsubscribed on SMS channel does not change their subscription status

The import process can recognize existing recipients, using the mobile number as unique identifiers, and update them according to the given input details. It is possible to add some query string parameters to specify some advanced behavior for the import process:

- **clearMissingOrEmptyFields:** when set to "true", the import process clears on the MailUp field that is not specified or set as empty in the request body.
- **removeFromPrevGroups:** when set to "true", first the import process removes the given recipients from any group they have been part of, then it adds them only to the group specified along with this request. This query string parameter applies only to "Add to a MailUp group" cases.

Description	Asynchronous import of one or more recipients on SMS channel
HTTP Method	POST
URL	<div><p>Add to a MailUp list</p><pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{id_List}/Recipients</pre></div> <div><p>Add to a MailUp group</p><pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Group/{id_Group}/Recipients</pre></div>

Reference	<p>Go to automated doc</p> <p>The request is composed of a list of comma-separated objects, each one of them is composed of the following mandatory fields:</p> <table border="1" data-bbox="408 344 1437 1084"> <thead> <tr> <th>Field Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Fields</td> <td>Array of objects [0..39]. Each object is composed of the following fields: <ul style="list-style-type: none"> Description (String, optional) Id (Integer, from 1 to 39) Value (String) </td> <td>Each array object is a personal data field. If the recipient is already existing, the "fields" values can be used to update the recipient's data</td> </tr> <tr> <td>Name</td> <td>String</td> <td>Not used but required, it can be empty (i.e. "Name": "")</td> </tr> <tr> <td>MobileNumber</td> <td>String</td> <td>The number format is checked by an asynchronous process, hence in case of invalid number import fails but no error is returned</td> </tr> <tr> <td>MobilePrefix</td> <td>String</td> <td>International prefix. Allowed formats are: <ul style="list-style-type: none"> prefix with 00 (e.g. "0039") prefix with + (e.g. "+39") empty prefix (e.g. "", default list prefix is applied) </td> </tr> </tbody> </table>	Field Name	Type	Description	Fields	Array of objects [0..39]. Each object is composed of the following fields: <ul style="list-style-type: none"> Description (String, optional) Id (Integer, from 1 to 39) Value (String) 	Each array object is a personal data field. If the recipient is already existing, the "fields" values can be used to update the recipient's data	Name	String	Not used but required, it can be empty (i.e. "Name": "")	MobileNumber	String	The number format is checked by an asynchronous process, hence in case of invalid number import fails but no error is returned	MobilePrefix	String	International prefix. Allowed formats are: <ul style="list-style-type: none"> prefix with 00 (e.g. "0039") prefix with + (e.g. "+39") empty prefix (e.g. "", default list prefix is applied)
Field Name	Type	Description														
Fields	Array of objects [0..39]. Each object is composed of the following fields: <ul style="list-style-type: none"> Description (String, optional) Id (Integer, from 1 to 39) Value (String) 	Each array object is a personal data field. If the recipient is already existing, the "fields" values can be used to update the recipient's data														
Name	String	Not used but required, it can be empty (i.e. "Name": "")														
MobileNumber	String	The number format is checked by an asynchronous process, hence in case of invalid number import fails but no error is returned														
MobilePrefix	String	International prefix. Allowed formats are: <ul style="list-style-type: none"> prefix with 00 (e.g. "0039") prefix with + (e.g. "+39") empty prefix (e.g. "", default list prefix is applied) 														
JSON request (example)	<pre>[{ "Fields": [{ "Description": "String content", "Id": 1, "Value": "String content" }], "Name": "String content", "MobileNumber": "10123456", "MobilePrefix": "0039" }]</pre>															
JSON response (example)	Returns import ID															
Paging and filtering (example)																

Read recipients

Description	Specific methods are available for retrieving recipients that are, respectively, subscribed, pending, or unsubscribed on the SMS channel.
HTTP Method	GET

URL	<pre> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/ {id_List}/Recipients/Subscribed https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/ {id_List}/Recipients/Pending https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/ {id_List}/Recipients/Unsubscribed </pre>
Reference	Automated docs are available for Subscribed , Pending , and Unsubscribed
JSON request (example)	
JSON response (example)	<p>The same data format is used in the responses related to "Subscribed", "Pending" and "Unsubscribed". Full recipient details are returned.</p> <pre> { "IsPaginated" : false, "Items" : [{ "Fields" : [{ "Description" : "LatestOrderAmount", "Id" : 15, "Value" : "" }, { "Description" : "FirstName", "Id" : 1, "Value" : "Mary" }, { "Description" : "Region", "Id" : 8, "Value" : "" }, { "Description" : "LatestOrderProductIDs", "Id" : 16, "Value" : "" }, { "Description" : "LastName", "Id" : 2, "Value" : "Johnson" }, { "Description" : "Address", "Id" : 9, "Value" : "" }, { "Description" : "LatestOrderCategoryIDs", "Id" : 17, "Value" : "" }, { "Description" : "Company", "Id" : 3, "Value" : "" }, { "Description" : "Gender ", "Id" : 10, "Value" : "" }] }] } </pre>

```
},
{
  "Description" : "LatestShippedOrderDate",
  "Id" : 18,
  "Value" : ""
},
{
  "Description" : "City",
  "Id" : 4,
  "Value" : ""
},
{
  "Description" : "phone",
  "Id" : 11,
  "Value" : ""
},
{
  "Description" : "LatestShippedOrderID",
  "Id" : 19,
  "Value" : ""
},
{
  "Description" : "Province",
  "Id" : 5,
  "Value" : ""
},
{
  "Description" : "CustomerID",
  "Id" : 12,
  "Value" : ""
},
{
  "Description" : "LatestAbandonedCartDate",
  "Id" : 20,
  "Value" : ""
},
{
  "Description" : "ZIP",
  "Id" : 6,
  "Value" : ""
},
{
  "Description" : "LatestOrderID",
  "Id" : 13,
  "Value" : ""
},
{
  "Description" : "State",
  "Id" : 7,
  "Value" : ""
},
{
  "Description" : "LatestOrderDate",
  "Id" : 14,
  "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" : ""
    }
    ],
    "Name" : null,
    "idRecipient" : 15,
    "Email" : null,
    "MobileNumber" : "323456789",
    "MobilePrefix" : "001"
    }
    ],
    "PageNumber" : 0,
    "PageSize" : 10,
    "Skipped" : 0,
    "TotalElementsCount" : 1
    }

```

Paging and filtering (example)

Text messages

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

Create a message

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

Known limitations:

- In some countries, like Italy, some restrictions on SMS senders apply. Use MailUp admin console to verify if the specified sender is subject to limitations in some countries
- MailUp REST API applies, on alphanumeric senders, looser format restrictions than the admin console. While the admin console allows only letters and numbers, API also allow spaces, punctuation, and other special characters. Please consider that letters and numbers are supported by any carrier, while other characters may be replaced in some cases. It is recommended that you perform some tests before using a sender with spaces, punctuation, or special characters
- In the case of an empty Sender, the default list settings are applied

Description	Create a new text message
HTTP Method	POST
URL	<pre> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{id_List}/Message </pre>

Reference[Go to automated doc](#)

Field Name	Type	Description
Subject	String	Message title (for MailUp users only, it will not be visible to recipients)
idList	Integer	List-ID
Content	String	Message content
Fields	Array of objects [0..39]. Each object is composed of the following fields: <ul style="list-style-type: none">• Description (String, optional)• Id (Integer, from 1 to 39)• Value (String)	Each array object will be used to fill the correspondent dynamic field when its value has been left empty or blank.
Notes	String	Message notes (for MailUp users only, it will not be visible to recipients)
Tags	Array of objects Each object is composed of the following fields: <ul style="list-style-type: none">• Enabled (Boolean)• Id (Integer, from 1 to 39)• Value (String)	An optional field is used to link existing tags to the text message. Tags can be used to classify messages, please check this page for more details about tags
IsUnicode	Boolean	False if the message contains only letters or symbols that belong to the Western alphabet (GSM charset). True if at least one message character belongs to a non-western alphabet (Unicode)
Sender	String	Message sender. It can be a phone number (e.g. +393351234567) or a string (up to 11 chars, only letters, and numbers). Please note that in some countries, like Italy, some restrictions on SMS senders apply. Use the MailUp admin console to verify if the specified sender is subject to limitations in some countries. If you leave this field empty will be applied the SMS sender defined for the current List.

JSON request (example)	<p>In this case recipients with an empty field as "FirstName" will receive "Hello dear customer, your appointment is confirmed for today at 5 PM, Main Building 1st floor" from "ABCandCO"</p> <pre> { "Subject": "Appointment #1", "idList": 2, "Content": "Hello [firstname], your appointment is confirmed for today at 5PM, Main Building 1st floor", "Fields": [{ "Id": 1, "Value": "dear customer" }], "Notes": "Appointment is confirmed", "IsUnicode": false, "Sender": "ABCandCO" } </pre>
JSON response (example)	<pre> { "Subject" : "Appointment #1", "idList" : 1, "idMessage" : 2 } </pre>
Paging and filtering (example)	

List Messages

Description	Retrieve text messages of a specified list
HTTP Method	GET
URL	<pre> GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{id_List}/Messages </pre>
Reference	Go to automated doc
JSON request (example)	none

JSON response (example)	<p>All the text messages of a specified list</p> <pre> { "IsPaginated" : false, "Items" : [{ "Subject" : "Appointment #1", "idList" : 2, "idMessage" : 2 }, { "Subject" : "Happy Holidays", "idList" : 2, "idMessage" : 6 }, { "Subject" : "Happy Birthday", "idList" : 2, "idMessage" : 5 }], "PageNumber" : 0, "PageSize" : 10, "Skipped" : 0, "TotalElementsCount" : 3 } </pre>
Paging and filtering (example)	<p>Use this filter to extract only the SMS messages of list 1 that contain the word "promo" in their subject. Strict filter matching with "==" instead of "Contains" is much more performing.</p> <pre> GET /Console/Sms/List/1/Messages?filterby="Subject.Contains('promo')" </pre>

Read message details

Description	Retrieve the text message details by specified message-id
HTTP Method	GET
URL	Retrieve the SMS details by specified id <pre> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{id_List}/Message/{id_Message} </pre>
Reference	Go to automated doc
JSON request (example)	none
JSON response (example)	<pre> { "Subject" : "Appointment #1", "idList" : 1, "idMessage" : 2, "Content" : "Hello [firstname], your appointment is confirmed for today at 5PM, Main Building 1st floor", "Fields" : [{ "Description" : "FirstName", "Id" : 1, "Value" : "dear customer" }], } </pre>


```
{
  "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" : ""
    }
  ],
  "Notes" : "Appointment is confirmed",
  "Tags" : [

],
  "IsUnicode" : false,
  "Sender" : "ABCandCO"
}

```

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

Update message

Description	Update a text message by specifying the message-id
HTTP Method	PUT
URL	<pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{ID_LIST}/Message/{ID_MESSAGE}</pre>
Reference	Go to automated doc
JSON request (example)	<pre>{ "Subject" : "Appointment #1", "idList" : 1, "idMessage" : 2, "Content" : "Hello [firstname], your appointment is confirmed for today at 5PM, Main Building 1st floor", }</pre>
JSON response (example)	<pre>{ "Subject" : "Appointment #1", "idList" : 1, "idMessage" : 2, }</pre>
Paging and filtering (example)	

Delete Message

Description	Delete a text message by specifying its ID
HTTP Method	DELETE
URL	<pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{ID_LIST}/Message/{ID_MESSAGE}</pre>
Reference	Go to automated doc
JSON request (example)	none
JSON response (example)	none
Paging and filtering (example)	

Send a text message



SMS Sender

Regardless of the sender, you set when creating a message, MailUp REST API always sends SMS using the sender that is specified at the list level. If you want to change the sender you must modify the list properties. Please note that in some countries the sender may be subject to local restrictions and it may be overwritten by the carriers before being delivered. You can contact us if you need support for some specific nations.

Single Send

Description	Send a text message to a mobile number by specifying message ID. The recipient is automatically subscribed to the SMS channel of the list in which the message was created.
HTTP Method	POST
URL	<pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Send</pre>
Reference	Go to automated doc
JSON request (example)	<pre>{ "Number": "3351234567", "Prefix": "0039", "idMessage": 27 }</pre>
JSON response (example)	<pre>{"InvalidRecipients": [], "Sent": 1, "UnprocessedRecipients": [], "idMessage": 27}</pre>
Paging and filtering (example)	

Send to group/list

Description	Immediate or scheduled mailing to a list or a group
HTTP Method	POST

URL	<pre><!-- Immediate Send to all subscribers on SMS channel for specified list --> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{id_List} /Message/{id_Message}/Send <!-- Immediate Send to all subscribers on SMS channel for specified group --> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Group/ {id_Group}/Message/{id_Message}/Send <!-- Scheduled Send to all subscribers on SMS channel for specified list (UTC timezone is expected) --> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/List/{id_List} /Message/{id_Message}/Send?datetime="2015-10-20 05:00:00Z" <!-- Scheduled Send to all subscribers on SMS channel for specified group (UTC timezone is expected) --> https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Group/ {id_Group}/Message/{id_Message}/Send?datetime="2015-10-20 05:00:00Z"</pre>
Reference	Go to automated doc (all operations)
JSON request (example)	none
JSON response (example)	<p>"Sent" value refers to the number of recipients that will receive the message regardless of the type of sending (immediate or deferred)</p> <pre>{"InvalidRecipients": [], "Sent": 10, "UnprocessedRecipients": [], "idMessage": 42}</pre>
Paging and filtering (example)	

Manage Send Queue

Known limitations:

- scheduled date and time cannot be modified
- at the moment there is no way to move from immediate/scheduled queues to "waiting to be sent" queue

Description	Query the send queue
HTTP Method	GET, POST

URL	<pre> <!-- Retrieve the status of sms sending queues for immediate scheduled dispatches --> GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Sendings/Immediate <!-- Retrieve the status of sms sending queues for deferred scheduled dispatches --> GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Sendings/Deferred <!-- Retrieves the status of sms sending queues for dispatches to be scheduled --> GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Sendings/Undefined <!-- Retrieves the earliest date and time to schedule the given sending task --> GET https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Sending/{id_Sending}/Deferred <!-- Move from "waiting to be sent" to "immediate sending" queue (without starting send) --> POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Sendings/{id_Sending}/Immediate <!-- Move from "waiting to be sent" to "scheduled sending" queue (scheduled date and time is passed as POST parameter) --> POST https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/Sendings/{id_Sending}/Deferred </pre>
Reference	Go to automated doc (all operations)
JSON request (example)	<p>Request example for POST /Console/Sms/Sendings/{id_Sending}/Deferred</p> <pre> { "Date": "2015-02-19 14:41:46Z" } </pre>

<p>JSON response (example)</p>	<div data-bbox="352 232 1434 427"> <p>Immediate</p> <pre>{ "IsPaginated":false,"Items":[{"Id":74,"IdList":10,"IdMessage":41,"Notes":"","StartDate":"","State":"Running","Subject":"Welcome","Type":"Immediate"}],"PageNumber":0,"PageSize":20,"Skipped":0,"TotalElementsCount":1}</pre> </div> <div data-bbox="352 450 1434 712"> <p>Deferred</p> <pre>{ "IsPaginated":false,"Items":[{"Id":72,"IdList":10,"IdMessage":29,"Notes":"This is a test message","StartDate":"2015-02-12 17:00:00Z","State":"Queued","Subject":"Test #1","Type":"Deferred"}, {"Id":73,"IdList":3,"IdMessage":19,"Notes":"","StartDate":"2015-03-02 15:00:00Z","State":"Queued","Subject":"Customer credentials","Type":"Deferred"}], "PageNumber":0,"PageSize":20,"Skipped":0,"TotalElementsCount":2}</pre> </div> <div data-bbox="352 734 1434 1160"> <p>Undefined</p> <pre>{ "IsPaginated":false,"Items":[{"Id":36,"IdList":8,"IdMessage":12,"Notes":"msg to be sent after 3 warnings","StartDate":null,"State":"ToBePlanned","Subject":"Final notice","Type":"Undefined"}, {"Id":44,"IdList":3,"IdMessage":4,"Notes":"","StartDate":null,"State":"ToBePlanned","Subject":"Valentine's day","Type":"Undefined"}, {"Id":46,"IdList":2,"IdMessage":14,"Notes":"","StartDate":null,"State":"ToBePlanned","Subject":"Welcome","Type":"Undefined"}, {"Id":52,"IdList":2,"IdMessage":18,"Notes":"","StartDate":null,"State":"ToBePlanned","Subject":"Happy birthday","Type":"Undefined"}, {"Id":48,"IdList":2,"IdMessage":15,"Notes":"","StartDate":null,"State":"ToBePlanned","Subject":"Season greetings","Type":"Undefined"}, {"Id":71,"IdList":10,"IdMessage":30,"Notes":"","StartDate":null,"State":"ToBePlanned","Subject":"Password reminder","Type":"Undefined"}], "PageNumber":0,"PageSize":20,"Skipped":0,"TotalElementsCount":6}</pre> </div> <div data-bbox="352 1182 1434 1317"> <p>Earliest date and time</p> <pre>"2015-02-09 14:41:46Z"</pre> </div> <div data-bbox="352 1339 1434 1503"> <p>Moved from waiting to immediate</p> <pre>{"Id":74,"IdList":10,"IdMessage":41,"Notes":"","StartDate":"","State":"Running","Subject":"Welcome","Type":"Immediate"}</pre> </div> <div data-bbox="352 1525 1434 1688"> <p>Moved from waiting to deferred</p> <pre>{"Id":74,"IdList":10,"IdMessage":41,"Notes":"","StartDate":"2015-02-19 14:41:46Z","State":"Queued","Subject":"Welcome","Type":"Deferred"}</pre> </div>
<p>Paging and filtering (example)</p>	

Summary report

Description	Retrieve the SMS message sending report by specifying SMS ID.
HTTP Method	GET
URL	<pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/{id_Message}/Sendings/Report</pre>
Reference	Go to automated doc
JSON request (example)	none
JSON response (example)	<p>StartDate = timestamp for mailing start EndDate = timestamp for mailing start Duration = mailing duration (from start to end dates) IsCompleted = true if mailing is finished TotalCost = number of spent credits</p> <p>TotalMessagesSent = number of messages that compose the mailing (a concatenated message is treated as a single message)</p> <p>TotalSmsSent = number of billed messages (a concatenated message is billed as a multiple message)</p> <pre>{ "Duration" : "00:00:06.4270000", "EndDate" : "2014-11-27 10:35:49Z", "IsCompleted" : true, "StartDate" : "2014-11-27 10:35:43Z", "TotalCost" : 2, "TotalMessagesSent" : 1, "TotalSmsSent" : 2 }</pre>
Paging and filtering (example)	

Detailed report

Description	Retrieve the SMS message sending report details by specifying SMS ID.
HTTP Method	GET
URL	<pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Sms/{ID_MESSAGE}/Sendings/ReportDetails</pre>
Reference	Go to automated doc
JSON request (example)	none

<p>JSON response (example)</p>	<p>ErrorCode = operation result (FAILED, UNKNOWN, NACK, UNDELIV, REJECTD, EXPIRED) + specific error code</p> <p>Receiver = recipient's phone number, international prefix is specified using neither '+' nor '00'</p> <p>SentDate = sent date and time (UTC time zone)</p> <p>Status = 0 (delivered), 1 (queued), 2 (error)</p> <pre> { "IsPaginated":false,"Items":[{ "ErrorCode":"FAILED_AT_004","Receiver":"391111111111","SentDate":"2015-01-29 14:11:10Z","Status":2}, { "ErrorCode":"FAILED_AT_004","Receiver":"392222222222","SentDate":"2015-01-29 14:11:10Z","Status":2}, { "ErrorCode":"FAILED_AT_004","Receiver":"393333333333","SentDate":"2015-01-29 14:11:10Z","Status":2}, { "ErrorCode":"FAILED_AT_004","Receiver":"394444444444","SentDate":"2015-01-29 14:11:10Z","Status":2}, { "ErrorCode":"FAILED_AT_004","Receiver":"395555555555","SentDate":"2015-01-29 14:11:10Z","Status":2}, { "ErrorCode":"","Receiver":"396666666666","SentDate":"2015-01-29 14:11:10Z","Status":0}, { "ErrorCode":"","Receiver":"397777777777","SentDate":"2015-01-29 14:11:10Z","Status":0}, { "ErrorCode":"FAILED_AT_004","Receiver":"398888888888","SentDate":"2015-01-29 14:11:10Z","Status":2}, { "ErrorCode":"","Receiver":"399999999999","SentDate":"2015-01-29 14:11:10Z","Status":0}], "PageNumber":0, "PageSize":0, "Skipped":0, "TotalElementsCount":0} </pre>
<p>Paging and filtering (example)</p>	<p>Use this filter to extract the result of the latest message with ID=24 that was sent to number +3933512345678 GET /Console/Sms/24/Sendings/ReportDetails?filterby="Receiver=='3933512345678'"&orderby="SentDate+desc"&pageSize=1&pageNumber=0</p> <p>If you used the sending method of the Transactional API for SMS you can also filter by the DeliveryId that was returned by that method.</p> <p>GET /Console/Sms/24/Sendings/ReportDetails?DeliveryId=1648</p> <p>In this case, paging and further filtering is not required since no more than one result can be returned</p>

SMS Credits

Summary report

Description	Retrieve several account details, including the number of available credits
HTTP Method	GET
URL	<pre>https://services.mailup.com/API/v1.1/Rest/ConsoleService.svc/Console/Authentication/Details</pre>
Reference	Go to automated doc
JSON request (example)	none

<p>JSON response (example)</p>	<p>Company = String, Customer name</p> <p>IsTrial = Boolean, true in case of trial account</p> <p>UID = String, user ID</p> <p>Username = String, user name</p> <p>Version = String, MailUp account version</p> <p>Email = String, address used to activate the MailUp account</p> <p>TotalCredits = Float, number of available credits</p> <p>CreditsByList = Array of objects, number of available credits for each list (if credit's splitting by list is enabled)</p> <p>Plan = Int, ID of purchased plan</p> <p>ExpiryDate = Date</p> <p>Autorenewal = Int, 1 if renewal is automatic, otherwise false</p> <p>Lang = String, language</p> <pre> { "Company": "String content", "IsTrial": true, "UID": "String content", "Username": "String content", "Version": "String content", "Email": "String content", "TotalCredits": 400.50 "CreditsByList": [{"ListId": 1, "Credit": 10.3}, {"ListId": 2, "Credit": 4.0}] "Plan": 1 "ExpiryDate": "2025-02-20 00:00:00.000" "AutoRenewal": 1 "Lang": "String content" } </pre>
<p>Paging and filtering (example)</p>	