

INTRODUCTION

Your software can be integrated with our gateway quickly and easily through REST API.

The REST API are based on HTTP methods and they can be integrated using a HTTP Client, based in any programming language.

Service Url

The REST API access url is:

<https://api.smstools.it/rest/api>

For security reasons, the REST APIs use only the HTTPS protocol (the HTTP protocol is not supported).

How to make a request

Authentication

All requests require authentication via the HTTP Basic authentication.

The username and password are the AUTH_KEY and AUTH_SECRET in your profile.

Credenziali REST API

```
</> AUTH_KEY : SSMHxxxxxxxxxxxxxxxxxxxx  
AUTH_SECRET : xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

These values can be retrieved in the API security management section in your private area. You can test your credentials performing a call with the following URL from your browser:

https://AUTH_KEY:AUTH_SECRET@api.smstools.it/rest/api/user

or using the linux shell command

```
curl --user AUTH KEY:AUTH SECRET https://api.smstools.it/rest/api/user
```

If authentication fails, a HTTP 401 error is returned.

Encoding & Content-Type

The requests must be encoded in UTF-8.

The value of the Content-Type header in POST and PUT Http Requests must be set to: `application/x-www-form-urlencoded`

Response Format

REST API can return response in both JSON (default) and XML format, based on parameter Accept specified in the HTTP request.

- **Accept:application/json** --> JSON response

```
curl -H "Accept: application/json" --user AUTH_KEY:AUTH_SECRET  
https://api.smstools.it/rest/api/user
```

- **Accept:application/xml** → XML response

```
curl -H "Accept: application/xml" --user AUTH_KEY:AUTH_SECRET  
https://api.smstools.it/rest/api/user
```

Sending SMS'es

- > Sending SMS'es
- > Sending bulk SMS
- > Sending binary SMS
- > Estimate cost of a text message
- > Cancel sending SMS

- > Search SMS
- > statusCallback Management
- > notificationCallback Management

Sending SMS'es

You can use this method to send a sms to one or more recipients or to one or more groups in the address book.

You can test the submission by setting the `sandbox` parameter to true in the request.

With the parameter `statusCallback` you can register a URL that can be used to send a notify via HTTP with SMS status (Delivered,Not Delivered).

For further information please see the follow table

URI

POST <https://api.smstools.it/rest/api/sms/send>

Request

For Post request the "**Content-Type**" must be "**application/x-www-form-urlencoded**"

Parameter	Mandatory	Type	Description
from	No	String	The sender of the SMS. Up to 11 alphanumeric characters, or 15 digits, that will be shown as the sender of the SMS. If not specified, the default sender is the sender specified in your profile.

to	No (it is Mandatory to evaluate at least one of the to and group fields)	String	SMS recipients. It must be a mobile number including an international prefix. You can specify multiple phone numbers (max 1000 numbers) by separating them with the comma character ",". For example: 393481234567 or 393481234567,393481234568,393481234568 Alternatively the parameter can be set with a JSON array containing the objects that describe the recipient. Every object must have compulsory msisdn appropriateness. The other properties of these objects can be used to parameterize the message text (see the text field) Example: [{msisdn:'393481234567'},{msisdn:'393481234568'},{msisdn:'393481234568'}]
----	--	--------	---

group	No (it is Mandatory to evaluate at least one of the to and group fields)	String	Group Id in address book It's possible to send SMS to more groups separating them with the comma character ",". For example if in the address book are present the groups "Amici"(id=1) and "Scuola"(id=2) you can specify: 1 or 1,2
text	Yes	String	The content of the SMS, always specified in UTF-8 encoding. If the message exceeds 160 characters, multiple chained text messages are sent up to a maximum of 5. The text may contain placeholders which will be replaced when sending the values associated with the respective recipient. The placeholder must be specified with the syntax \$ {property name} If the text contains placeholders, the to field must be filled in with a JSON array and each object in the array must contain the property indicated in the placeholder. For example: to=[{msisdn:'393481234567',nome:'Mario Rossi'},{msisdn:'393481234568',nome:'Luigi Bianchi'}] text=Invio sms di test a \${nome}

date	No	String	Sms scheduled Date (format yyyy-MM-ddTHH:mm:ssZ) to schedule message sending at certain time. If no value is specified, the message is sent immediately. The character Z indicates the timezone in the 4-digit RFC 822 format (For Italy +0200 during the summer schedule and +0100 in the remaining period of the year) Example: 2014-12-25T08:00:00+0200 indicates the date 25/12/2014 at 08:00:00 in GMT +2
transaction Id	No	String	A personal id to identificate a send. Allows you to assign all the text messages sent with a call the same identifier in order to use this parameter to receive the status of the messages. maximum 60 characters
sandbox	No	String	Allows you to test the call without actually sending the message. Use this parameter in the development enviroment Possible values true/false

statusCallb ack	No	String	If specified send status notifications to this URL
encoding	No	String	The encoding for the message. Possible values are: 7BIT - Encoding standard for GSM Message. Maximum length for a sms is 160 characters. UCS2 - Encoding a 8-bit to send GSM Message not ASCII. Maximum length for a sms is 70 characters. AUTO - The encoding to be used is automatically identified based on the content of the text field. Default : 7BIT

Response

HTTP Status Code

Code	Description
200	ok

400	Parameters not valid Possible errorMsg are: BAD_TRANSACTIONID - parameter transactionId not valid BAD_DATE - format date not valid BAD_TEXT - parameter text not valid BAD_TEXT_PLACEHOLDER - text contains non-replaceable placeholders BAD_CALLBACK - parameter statusCallback not valid BAD_ENCODING - parameter encoding not valid BAD_FROM - parameter from not valid BAD_GROUP - parameter group not valid BAD_CREDIT - insufficient credit NO_VALID_RECIPIENT - no valid recipients GENERIC_ERROR - internal server error
401	Invalid credentials
405	HTTP Method not allowed

500	Generic Error
-----	---------------

Sending Data

Property	Description
from	SMS sender
text	Sms Texty
transactionId	id sms If not specified a transactionId will be generated automatically by the API REST
smsInserted	Amount of Sms inserted
smsNotInserted	Amount of Sms not inserted
sms	Sms List Each sms is identified by the property: to - aka the full mobile phone number of the recipient. id - internal id of the message. You can use this value to request the status of the message with the Search SMS function status - SMS status. Possible values are INSERTED and NOT_INSERTED

	statusDetail - if the status is NOT_INSERTED this field contains the message error. Possible values are : BADNUMBERFORMAT - Incorrect recipient number DUPLICATESMS - SMS already entered in the last 24 hours with identical recipient, sender and text. BLACKLIST - recipient in blacklist.
--	---

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET -XPOST https://api.smstools.it/rest/api/sms/send \
-d "from=mrossi" \
-d "to=393480000000" \
-d "text=Test Sms"
```

```
{
  "from": "mrossi",
  "text": "Test Sms",
  "transactionId": "d41ebee38b6a04818aa707818d3161e",
  "smsInserted": 1,
  "smsNotInserted": 0,
  "sms": [
    {
      "id": 389310742,
      "to": "393412345678",
      "status": "INSERTED"
    }
  ]
}
```

Bulk SMS

Asynchronous SMS sending mode to be used with a large number of recipients (More than 1000 recipients per single request).

Once the request has been sent, a transactionId is returned, which is useful for subsequently identifying the sms.

If the sending cannot be completed due to an error, an email will be sent containing the error code that occurred

Nota: If you send up to 1000 recipients per single request, we advise you to use the SMS sending function

URI

POST <https://api.smstools.it/rest/api/sms/sendbulk>

Request

The parameters of the request are the same as for SMS sending

Additionally, bulk sending includes these parameters

Parameter	Mandatory	Type	Description
transactionCall back	No	String	If specified send status notifications to this URL.

Response

HTTP Status Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_TRANSACTIONID - parameter transactionId not valid BAD_DATE - date format not valid BAD_TEXT - parameter text not valid BAD_TEXT_PLACEHOLDER - text contains non-replaceable placeholders BAD_CALLBACK - parameter statusCallback not valid BAD_ENCODING - parameter encoding not valid BAD_FROM - parameter from not valid BAD_GROUP - parameter group not valid BAD_CREDIT - bad credit NO_VALID_RECIPIENT - no valid recipient GENERIC_ERROR - internal server error
401	Invalid credentials
405	HTTP Method not allowed

500	Generic error
-----	---------------

Sending Data

Property	Description
from	message sender.
text	Sms Text
transaction Id	message ids If not specified a transactionId will be generated automatically by the REST API

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET -XPOST https://api.smstools.it/rest/api/sms/sendbulk \
-d "from=mrossi" \
-d "to=393480000000" \
-d "text=Test Sms"
```

```
{
  "from": "mrossi",
  "text": "Test Sms",
  "transactionId": "e51ea840fc7276922ac61b733027301d"
}
```

Sending a Binary SMS

Sending an SMS with binary content.

Notes on the length of the message:

The maximum length of a single binary SMS is 140 bytes (280 characters expressed in hexadecimal format)

By specifying a longer text, concatenated SMS will be sent. The space available for concatenated SMS is 134 bytes per SMS (268 characters expressed in hexadecimal format)

URI

POST <https://api.smstools.it/rest/api/sms/sendbinary>

Request

The parameters of the request are the same as for SMS sending with the following variations

parameter	Mandatory	Type	Description
text	Yes	String	Binary content of the message as a hexadecimal string. Each byte must be represented with a two-character hexadecimal value. Example: 0fc24abf3413ba
udh	No	String	Optional field containing the UDH (User Data Header) of the message in hexadecimal format. Example: 00c34abf300311
pid	No	Integer	Optional field containing the protocol id (Protocol identifier) to be used in decimal format.

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_TRANSACTIONID - parameter transactionId not valid BAD_DATE - format date not valid BAD_TEXT - parameter text not valid BAD_TEXT_PLACEHOLDER - text contains non-replaceable placeholders BAD_CALLBACK - parameter statusCallback not valid BAD_ENCODING - parameter encoding not valid BAD_FROM - parameter from not valid BAD_GROUP - parameter group not valid BAD_CREDIT - No Credit NO_VALID_RECIPIENT - no valid recipients

	BAD_TEXT_HEX - the text does not contain a valid hexadecimal BAD_UDH_HEX - udh does not contain a valid hexadecimal BAD_PID - invalid pid GENERIC_ERROR - internal server error
401	Invalid credentials
405	HTTP Method not allowed
500	Generic error

Sending Data

The return data sent is the same as for sending an SMS

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET -XPOST https://api.smstools.it/rest/api/sms/sendbinary \
-d "from=mrossi" \
-d "to=393480000000" \
-d "text=0fc24abf3413ba"
```



```
{
  "from": "mrossi",
  "text": "0fc24abf3413ba",
  "transactionId": "e51ea840fc7276922ac61b733027301d"
}
```

SMS estimate cost

URI

POST <https://api.smstools.it/rest/api/sms/estimate>

Request

"Content-Type" for request must be "application/x-www-form-urlencoded"

parameter	Mandatory	Type	Description
from	No	String	The sender of the SMS. The originator of the SMS message. The SMS standards limit the length up to 15 digits if it's a number and up to 11 characters if it's a text. You can use spaces in the sender, but most modern smartphones do not display the space.

to	No (Mandatory at least one: to or group)	String	The recipient of the message. It must be a mobile number including an international prefix. You can specify multiple phone numbers (max 1000 numbers) separating them with the comma character "," . Example: 393481234567 or 393481234567,393481234568,393481234568
group	No (Mandatory at least one: to or group)	String	Id Group in address book You can specify multiple groups in the address book by separating them with the comma character ",". Example for two groups "Amici"(id=1) and "Scuola"(id=2) it will be possible to indicate: 1 or 1,2
text	Yes	String	Message Text

encoding	No	String	Sms encoding Possible values are: 7BIT - Standard encoding for GSM messages. The maximum length for a single text message is 160 characters. UCS2 - 8-bit encoding to send GSM messages with non-ASCII content. The maximum length for a single text message is 70 characters. AUTO - The encoding to be used is automatically identified based on the content of the text field. Default : 7BIT
----------	----	--------	--

Response

HTTP Status Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_TEXT - parameter text not valid BAD_ENCODING - parameter encoding not valid BAD_FROM - parameter from not valid BAD_GROUP - parameter group not valid NO_VALID_RECIPIENT - no valid recipients GENERIC_ERROR - internal server error
401	Invalid credentials
405	HTTP Method not allowed
500	Generic error

Data of estimate cost

Property	Description
cost	Cost
userCredit	Credit amount

smsCount	SMS amount
----------	------------

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET -XPOST https://api.smstools.it/rest/api/sms/estimate \
-d "from=mrossi" \
-d "to=393480000000" \
-d "text=Test Sms"
```

```
{
  "cost":0.095,
  "userCredit":15.484,
  "smsCount":2
}
```

Cancel SMS send

You can cancel a sms not yet sent.

Use this method to cancel a scheduled sms

URI

POST <https://api.smstools.it/rest/api/sms/cancel>

Request

"Content-Type" for **POST** request must be "application/x-www-form-urlencoded"

parameter	Mandatory	Type	Description
-----------	-----------	------	-------------

id	No *	String	Id sms You can specify multiple ids separating them with the comma character ",". Example: 12312 or 432432,543543,435435
transactionId	No *	String	TransactionId assigned to the send

*** It is required that at least one parameter is set**

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: NO_PARAMS - No parameter set BAD_ID - format id not valid
401	Invalid credentials

405	HTTP Method not allowed
500	Generic error

The call returns the list of canceled text messages.

For the details of the Property of each single sms see the Search SMS function.

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET -XPOST https://api.smstools.it/rest/api/sms/cancel \
-d "id=389451366"
```

```
[
  {
    "id":389451366,
    "to":"393480000000",
    "text":"Test Sms",
    "from":"mrossi",
    "status":"NOSENT",
    "insertDate":"2014-01-16T13:46:11+0100",
    "transactionId":"1389876345449",
    "price":0.0
  }
]
```

Search SMS

Search for sent text messages.

URI

GET <https://api.smstools.it/rest/api/sms/search>

Request

parameter	Mandatory	Type	Description
id	No *	String	Internal ID of the message sent You can specify multiple ids separating them with the comma character " , " . Example: 12312 or 432432,543543,435435
transactionId	No *	String	TransactionId assigned to sending
msisdn	No *	String	sms number sent
fromDate	No *	String	Date from. format yyyy-MM-ddTHH:mm:ssZ
toDate	No	String	Date To. format yyyy-MM-ddTHH:mm:ssZ
status	No *	String	State of the sms

offset	No	Integer	Position of the first result that will be returned. Example: If the search finds 100 sms, the offset field is set to 50 and the limit field to 10 will be returned the sms from 51 to 60. Default : 0
limit	No	Integer	Number of results Default : 20

*** It is required that at least one parameter is set**

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: NO_PARAMS - No parameter set BAD_ID - parameter id not valid

	BAD_MSISDN - parameter msisdn not valid BAD_DATE - Format date not valid BAD_STATUS - parameter status not valid
401	Invalid credentials
405	HTTP Method not allowed
500	Generic error

The call returns a list of sent text messages.

Property	Description
smsList	List of received SMS Each sms is described by the Property: id - sms id to - recipient of the message text - sms text from - sms sender

	status - status of the message. See the table below. insertDate - date entry of the message in the format yyyy-MM-ddTHH:mm:ssZ sentDate - sent date in the format yyyy-MM-ddTHH:mm:ssZ deliveryDate - delivery date in the format yyyy-MM-ddTHH:mm:ssZ transactionId - sending id price - sms cost
metadata	Metadata For more details please visit this page

Sms status values:

Code	Description
PENDING	Sms in the queue
SENT	Sms sent to operator
NOSENT	Sms not sent to operator
DELIVERED	SMS delivered to the recipient's mobile phone

NOTDELIVERED	SMS not delivered to the recipient's mobile phone
--------------	---

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/sms/search?fromDate=2011-01-01T00:00:00-0200
```

```
{
  "metadata":{
    "count":1,
    "offset":0,
    "limit":20
  },
  "smsList":[
    {
      "id":389395385,
      "to":"393480000000",
      "text":"Test Sms",
      "from":"mrossi",
      "status":"DELIVERED",
      "insertDate":"2014-01-13T15:01:35+0100",
      "sentDate":"2014-01-13T15:01:37+0100",
      "deliveryDate":"2014-01-13T15:01:40+0100",
      "transactionId":"72810108d4114097b9edb42b255b273c",
      "price":0.0945
    }
  ]
}
```

Manage statusCallback

How to manage statusCallback calls sent by the platform to the URL indicated in the sending phase.

If you do not want to manage the Callback status, you can at any time query the status of one or more messages sent with the search SMS function.

URI

POST `http(s)://yourcallbackurl`

parameter	Type	Description
id	String	Internal ID of the message sent
transactionId	String	personal id sms
to	String	recipient of the message.
status	String	message status. Values are DELIVERED or NOTDELIVERED.
notification Date	String	format yyyy-MM-ddTHH:mm:ssZ

Response

HTTP Status Code

Code	Description
200	ok.
other Code	A new attempt is made to send the status up to a maximum of 20 times.

Status code

Code	Description
PENDING	SMS queued for sending
SENT	Sms Sent to operator
NOSENT	Sms not sent to operator
DELIVERED	SMS delivered to the recipient's mobile phone
NOTDELIVERED	SMS not delivered to the recipient's mobile phone

Manage notificationCallback

In bulk submission it is possible to specify a URL as parameter notificationCallback.

Calls will be made in POST with application-json content-type and will contain a json described in the following table

URI

POST http(s)://yourcallbackurl

parameter	Type	Description
transactionId	String	personal id sms

status	String	sms status Possible values are: ERROR - sms not sent general error. For more details see errorMsg parameter COMPLETED - sms sent
--------	--------	--

description (Status **ERROR**)

parameter	Type	Description
errorMsg	String	Possible errorMsg are: BAD_TEXT_PLACEHOLDER - ext contains non-replaceable placeholders NO_CREDIT - No Credit NO_VALID_RECIPIENT - no valid recipients GENERIC_ERROR - internal server error

Example

```
{
  "status": "ERROR",
  "errorMsg": "NO_CREDIT",
  "transactionId": "1ccf2c63472387ede0dbbfef9996d2d4"
```

}

Description (status **COMPLETED**)

Property	Description
from	message sender.
text	Sms Text
transactionId	sending id If not specified a transactionId will be generated automatically by the REST API
smsInserted	SMS number inserted
smsNotInserted	SMS number not inserted
sms	SMS List Each sms is described by the Property: to - sms recipient id - sms id status - values are INSERTED or NOT_INSERTED . statusDetail - message error in case of status NOT_INSERTED

	Possible values are : BADNUMBERFORMAT - recipient number not valid DUPLICATESMS - Sms already inserted BLACKLIST - recipient in blacklist.
--	---

Example Response Completed

```
{
  "status":"COMPLETED",
  "from":"Mario",
  "text":"test",
  "transactionId":"e57ad82669d5bf96ac37d18994374d5b",
  "smsInserted":0,
  "smsNotInserted":1,
  "sms":[
    {
      "to":"39340000000",
      "status":"INSERTED",
      "id":421420576
    }
  ]
}
```

Receiving SMS'es

- > Search for received SMS
- > List of reception services

Receiving SMS'es

Search for text messages received with the reception service "**Forward sms**".

URI

GET <https://api.smstools.it/rest/api/sms/received/search>

Request

parameter	Mandatory	Type	Description
from	No (Mandatory enhance at least one of the field from , simIdRef , fromDate e toDate)	String	Sender of the text messages received. It must be a mobile number including an international prefix. You can specify multiple phone numbers by separating them with the comma character ",". Example: 393481234567 or 393481234567,393481234568,393481234568
simIdRef	No (Mandatory enhance at least one of the field from, simIdRef, fromDate e toDate)	String	Identification of the reception service on which text messages were received
fromDate	No (Mandatory enhance at least one of the field from, simIdRef, fromDate e toDate)	String	date from of receipt, format yyyy-MM-ddTHH:mm:ssZ

toDate	No (Mandatory enhance at least one of the field from, simIdRef, fromDate e toDate)	String	date to of receipt, format yyyy-MM-ddTHH:mm:ssZ
offset	No	Integer	Position of the first result that will be returned. Example: If the search finds 100 sms, the offset field is set to 50 and the limit field to 10 will be returned the sms from 51 to 60. Default : 0
limit	No	Integer	Number of results Default : 20

Response

HTTP Status Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: NO_PARAMS - parameter mancante BAD_DATE - Wrong Format date
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

The call returns a list of received text messages

Property	Description
smsList	List of received SMS Each sms is described by the Property: from - sms sender text - sms text receiveDate - date reception of the message in the format yyyy-MM-ddTHH:mm:ssZ simId - id of the reception service on which the SMS was received

metadata	Metadata For more information see this page
----------	---

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/sms/received/search?fromDate=2011-01-01T00:00:00%2B0200
```

```
{
  "metadata":{
    "count":2,
    "offset":0,
    "limit":20
  },
  "smsList":[
    {
      "from":"+393450000000",
      "text":"Test sms 1",
      "receiveDate":"2011-04-21T13:23:13+0200"
    },
    {
      "from":"+393480000000",
      "text":"Test sms 1",
      "receiveDate":"2011-01-18T11:42:51+0100"
    }
  ]
}
```

List of reception Services

URI

GET <https://api.smstools.it/rest/api/sms/received/sim/list>

Request

parameter	Mandatory	Type	Description
No parameter required			

Response

HTTP Status Code

Code	Description
200	ok
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

The call returns a list of purchased services. Each service is described by the following Property

Property	Description
id	Reception service identification (Can be used in the search function to filter text messages)
sim	Number on which the reception service is active
key	If the reception service is shared, the key field contains the configured key

expirationDate	Service expiration date
----------------	-------------------------

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/sms/received/sim/list
```

```
[
  {
    "id": "265",
    "sim": "3933900000000",
    "key": "test",
    "receiveDate": "2011-04-21T13:23:13+0200"
  }
]
```

User

Information User

Returns the registry and credit related to your profile

URI

GET <https://api.smstools.it/rest/api/user>

Request

parameter	Mandatory		Description
No parameter required			

Response

HTTP Status Code

Code	Description
200	ok
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

User Data

Property	Description
name	Name
lastname	Surname
username	Account Username
email	Email
businessname	Business Name
address	Address
city	City

postcode	Postal Code
province	Province
country	Country
birthdate	date of birth
phone	Phone
msisdn	Mobile Station International Subscriber Directory Number, but you may think of this as the full mobile number
sender	Default Sender
senderAlias	Sender Alias
taxcode	Tax Code
vatnumber	Vat Number
registrationDate	registration date (format yyyy-MM-ddTHH:mm:ssZ)
expirationDate	Account expiration date in the format yyyy-MM-ddTHH:mm:ssZ
credit	Credit

italysms	Number of sms sent to Italy
----------	-----------------------------

Example:

CURL

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/user
```

```
{
  "name": "Mario",
  "lastname": "Rossi",
  "username": "mrossi",
  "email": "mario.rossi@fake.it",
  "address": "via Italia 1",
  "city": "Roma",
  "postcode": "00144",
  "province": "RM",
  "country": "IT",
  "birthdate": "01.01.1970",
  "phone": "060000000",
  "msisdn": "3934800000000",
  "sender": "mrossi",
  "taxcode": "MRXXXXXXXXXX",
  "registrationDate": 200607191808+0200,
  "expirationDate": 201408211148+0200,
  "credit": 15.693,
  "italysms": 166,
  "senderAlias":
  [
    "3934700000000",
    "mySender"
  ]
}
```

Phonergroup books

- > Group list
- > Group details
- > Group contacts
- > New group
- > refresh group
- > Delete group

Group list

List of groups in the phonebook

URI

GET <https://api.smstools.it/rest/api/phonebook/group/list>

Request

parameter	Mandatory	Type	Description
No parameter			

Response

HTTP Status Code

Code	Description
------	-------------

200	ok
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

The call returns a list with the groups in the phonebook.

Each group is described by the following properties:

Property	Description
id	Group id
name	Group Name
contactCount	Number of contacts in the group

Example:

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/phonebook/group/list
```

```
[
  {
    "id":1947,
    "name":"Amici",
    "contactCount":1
  },
  {
    "id":13197,
    "name":"Calcetto",
    "contactCount":1
  }
]
```

Group detail

URI

GET <https://api.smstools.it/rest/api/phonebook/group/{id}>

Request

parameter	Mandatory	Type	Description
id	Yes (parameter specified in the URI)	String	group id

Response

HTTP Status Code

Code	Description
200	ok
401	Invalid credentials
404	Group not found
405	HTTP Method not allowed
500	Generic Error

Property group:

Property	Description
----------	-------------

id	group id
name	group name
contactCount	Number of contacts in the group

Example:

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/phonebook/group/1947
```

```
{
  "id":1947,
  "name":"Amici",
  "contactCount":1
}
```

Group contacts

List of group contacts

URI

GET <https://api.smstools.it/rest/api/phonebook/group/{id}/contacts>

Request

parameter	Mandatory	Type	Description
id	Yes (specify it as URI parameter)	Integer	Group ID

offset	No	Integer	Position of the first result that will be returned. Example: If the search finds 100 contacts, the offset field is set to 50 and the limit field to 10 contacts from 51 to 60 will be returned. Default : 0
limit	No	Integer	Number of results Default : 20

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: NO_PARAMS - parameter name not specified
401	Invalid credentials

405	HTTP Method not allowed
500	Generic Error

The call returns a list of contacts in the phonebook.

Property	Description
contacts	List of contacts For the details of the Property of each contact, see the function “Contact Details”.
metadata	Metadata For more details please visit this page

Example:

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/phonebook/group/1947/contacts
```

```
{
  "metadata":{
    "count":1,
    "offset":0,
    "limit":20
  },
  "contacts":[
    {
      "msisdn":"393480000000",
      "name":"Mario",
      "lastname":"Rossi",
      "address":"via Italia 1",
      "city":"Roma",
      "postCode":00144,
      "province":"RM",
      "country":"IT",
      "email":"mario.rossi@fake.it",
      "homePhoneNumber":""
    }
  ]
}
```



```
{
  "groups":[
    {
      "id":13197,
      "name":"Amici"
    },
    {
      "id":1947,
      "name":"Lavoro"
    }
  ]
}
```

New group

Add a group

URI

POST <https://api.smstools.it/rest/api/phonebook/group>

Request

Il **"Content-Type"** per le richieste **POST** deve essere **"application/x-www-form-urlencoded"**

parameter	Mandatory	Type	Description
name	Yes	String	Group Name

Response

HTTP Status Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_GROUP_NAME - Group Name not valid GROUP_ALREADY_EXISTS - Already Exists
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

If the group is created correctly (HTTP status 200) the detail of the group is :

Property	Description
id	Group ID
name	Group Name
contactCount	Number of contacts in the group

Example:

```
curl --user AUTH_KEY:AUTH_SECRET -XPOST https://api.smstools.it/rest/api/phonebook/group \
-d "name=Amici"
```

```
{
  "id":1947,
  "name":"Amici",
  "contactCount":0
}
```

Update a Group

Update a Group

URI

PUT <https://api.smstools.it/rest/api/phonebook/group/{id}>

Request

the "**Content-Type**" for the **POST** request should be "**application/x-www-form-urlencoded**"

parameter	Mandatory	Type	Description
id	Yes (parameter URI)	Integer	Group ID
name	Yes	String	Group Name

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_GROUP_NAME - Group Name not valid

	GROUP_ALREADY_EXISTS - Group already exists
401	Invalid credentials
404	Group Not Found
405	HTTP Method not allowed
500	Generic Error

If the group is created correctly, the status of HTTP is 200

Example:

```
curl --user AUTH_KEY:AUTH_SECRET -XPUT https://api.smstools.it/rest/api/phonebook/group/1947 \
-d "name=Amici Calchetto"
```

HTTP status 200

Delete a Group

Delete a Group

URI

DELETE <https://api.smstools.it/rest/api/phonebook/group/{id}>

Request

parameter	Mandatory	Type	Description

id	Yes (parameter in the URI)	Integer	Group ID
delete_contacts	No	Boolean	Possible values:true/false If set to true all contacts that belong only to this group will be deleted. Default: false

Response

HTTP Status Code

Code	Description
204	ok
401	Invalid credentials
404	Group Not Found
405	HTTP Method not allowed
500	Generic Error

If the group is deleted correctly, the status “HTTP status 200” is returned

Example:

```
curl --user AUTH_KEY:AUTH_SECRET -XDELETE
https://api.smstools.it/rest/api/phonebook/group/1947?delete_contacts=true
```

HTTP status 204

Contatti Rubrica

- > Search Contacts
- > Contact Details
- > New Contact
- > Add contacts
- > **Add contact status**
- > Refresh Contact
- > Delete Contact

Search Contacts

URI

GET <https://api.smstools.it/rest/api/phonebook/contact/search>

Request

parameter	Mandatory	Type	Description
name	No	String	Name The search is carried out on the fields name e lastname
offset	No	Integer	Position of the first result that will be returned. Example: If the search finds 100 contacts, the offset field is set to 50 and the limit field to 10 contacts from 51 to 60 will be returned. Default : 0
limit	No	Integer	Number of results Default : 20

Se nella richiesta non viene specificato alcun parametro il servizio restituirà un elenco di tutti i contatti presenti in rubrica.

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: NO_PARAMS - parameter name not found
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

The call returns a list with contacts in the phonebook.

Property	Description
contacts	List of contacts For the details of the Property of each contact, see the function Contact detail.
metadata	Metadata For more details please visit this page

Example:


```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/phonebook/contact/search?name=Mario
```

```
{
  "metadata":{
    "count":1,
    "offset":0,
    "limit":20
  },
  "contacts":[
    {
      "msisdn":"393480000000",
      "name":"Mario",
      "lastname":"Rossi",
      "email":"mario.rossi@fake.it",
      "groups":[
        {
          "id":13197,
          "name":"Amici"
        },
        {
          "id":1947,
          "name":"Lavoro"
        }
      ],
      "customFields":[
        {
          "fieldKey": "f_indirizzo",
          "fieldName": "Indirizzo",
          "fieldValue": "via roma 1"
        },
        {
          "fieldKey": "f_cap",
          "fieldName": "CAP",
          "fieldValue": "30000"
        }
      ]
    }
  ]
}
```

Contact detail

URI

GET <https://api.smstools.it/rest/api/phonebook/contact/{msisdn}>

Request

parameter	Mandatory	Type	Description
msisdn	Yes (parameter in the URI)	String	Mobile Station International Subscriber Directory Number, but you may think of this as the full mobile number, including area code if available and the country code, but without prefixed zeros or '+'.

Response

HTTP Status Code

Code	Description
200	ok
401	Invalid credentials
404	Contact not found
405	HTTP Method not allowed
500	Generic Error

Property contact:

Property	Description

msisdn	Numero di cellulare del contatto
name	Nome del contatto
lastname	Cogome del contatto
email	Indirizzo Email
customFields	Lista dei campi personalizzati che sono stati valorizzati per il contatto Per ogni campo viene indicato il nome, la chiave ed il valore corrente
groups	Lista dei gruppi a cui il contatto appartiene. Per ogni gruppo viene indicato l'id interno ed il campo name

Example:

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/phonebook/contact/393480000000
```

```
{
  "msisdn": "393480000000",
  "name": "Mario",
  "lastname": "Rossi",
  "email": "mario.rossi@fake.it",
  "groups": [
    {
      "id": 13197,
      "name": "Amici"
    },
    {
      "id": 1947,
      "name": "Lavoro"
    }
  ],
  "customFields": [
```

```

{
  "fieldKey": "f_indirizzo",
  "fieldName": "Indirizzo",
  "fieldValue": "via roma 1"
},
{
  "fieldKey": "f_cap",
  "fieldName": "CAP",
  "fieldValue": "30000"
}
]
}

```

Nuovo contatto

Aggiungere contatto in rubrica

URI

POST <https://api.smstools.it/rest/api/phonebook/contact>

Request

Il **"Content-Type"** per le richieste **POST** deve essere **"application/x-www-form-urlencoded"**

parameter	Mandatory	Type	Description
msisdn	Yes	String	Numero di cellulare del contatto compreso di prefisso internazionale
name	Yes	String	Nome del contatto (max 20 caratteri)
lastname	No	String	Cognome del contatto (max: 20 caratteri)

email	No	String	Indirizzo Email
\${fieldKey}	No	String	Chiave/valore del campo personalizzato che si vuole popolare L'elenco delle chiavi disponibili, da usare come nome parameter, può essere recuperato tramite il servizio lista campi personalizzati
groupsId	No	String	Id interno del gruppo ai quale assegnare il contatto. È possibile specificare più gruppi separandoli con il carattere virgola ",". Ad Example: 1245 or 1245,3432

Response

HTTP Status Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_CONTACT_NAME - parameter name mancante BAD_CONTACT_MSISDN - parameter msisdn mancante o not valid BAD_CONTACT_COUNTRY - Code nazione not valid BAD_CONTACT_GROUP - Gruppo not valid CONTACT_ALREADY_EXISTS - Esiste già un contatto con questo msisdn
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

In caso il contatto venga creato correttamente (HTTP status 200) viene restituito il dettaglio del contatto

Per il dettaglio delle Property del contatto vedi la funzione [Contact detail](#).

Example:

```
curl --user AUTH_KEY:AUTH_SECRET -XPOST https://api.smstools.it/rest/api/phonebook/contact \
-d "msisdn=393480000000" \
-d "name=Mario" \
-d "lastname=Rossi" \
-d "f_indirizzo=via roma 1"
```

```
{
  "msisdn": "393480000000",
  "name": "Mario",
  "lastname": "Rossi"
```

```
{
  "customFields": [
    {
      "fieldKey": "f_indirizzo",
      "fieldName": "Indirizzo",
      "fieldValue": "via roma 1"
    }
  ]
}
```

Aggiungi contatti

Carica massivamente contatti.

URI

GET <https://api.smstools.it/rest/api/phonebook/contact/addContactBulk>

Request

Il **"Content-Type"** per le richieste **POST** deve essere **"multipart/form-data"**

parameter	Mandatory	Type	Description
file	Yes	File	File csv contenente i contatti da caricare. Il formato del file è identico all'import via web. Maggiori informazioni...

Response

HTTP Status Code

Code	Description
200	ok

401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Response

Property	Description
status	SUBMITTED File formalmente corretto. Elaborazione in corso REJECT File non corretto. Nessuna elaborazione BUSY C'è già una elaborazione in corso. Attendere il termine. (Vedi metodo successivo)
error	Description errore

Example:

CURL

JAVA

PHP

PYTHON

C#

```
curl --user AUTH_KEY:AUTH_SECRET \
-XPOST "https://api.smstools.it/rest/api/phonebook/contact/addContactBulk" \
-F "file=@prova_rubrica.csv"
```

```
{
  "status": "SUBMITTED"
}
```


Aggiungi contatti stato

Controlla lo stato della coda inserimento massivo contatti.

URI

GET https://api.smstools.it/rest/api/phonebook/contact/addContactBulk_status

Request

parameter	Mandatory	Type	Description
-----------	-----------	------	-------------

Response

HTTP Status Code

Code	Description
200	ok
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Response

Property	Description
progress	Percentuale di progressione
free	disponibile per invio successivo

Example:

CURL

JAVA

PHP

PYTHON

C#

```
curl --user AUTH_KEY:AUTH_SECRET \  
-XGET "https://api.smstools.it/rest/api/phonebook/contact/addContactBulk_status"
```

```
{  
  "progress": "100%",  
  "free": true  
}
```

Aggiorna contatto

Modificare un contatto in rubrica

URI

PUT https://api.smstools.it/rest/api/phonebook/contact/{msisdn}

Request

Il "**Content-Type**" per le richieste **POST** deve essere "**application/x-www-form-urlencoded**"

parameter	Mandatory	Type	Description
msisdn	Yes (parameter in the URI)	String	Numero di cellulare del contatto compreso di prefisso internazionale
name	Yes	String	Nome del contatto (max 20 caratteri)

lastname	No	String	Cogome del contatto (max 20 caratteri)
email	No	String	Indirizzo Email
{{fieldKey}}	No	String	Chiave/valore del campo personalizzato che si vuole popolare L'elenco delle chiavi disponibili, da usare come nome parameter, può essere recuperato tramite il servizio lista campi personalizzati
groupsId	No	String	Id inteno del gruppo ai quale assegnare il contatto. È possibile specificare più gruppi separandoli con il carattere virgola ",". Ad Example: 1245 or 1245,3432

Response

HTTP Status Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_CONTACT_NAME - parameter name mancante BAD_CONTACT_MSISDN - parameter msisdn mancante o not valid BAD_CONTACT_COUNTRY - Code nazione not valid BAD_CONTACT_GROUP - Gruppo not valid CONTACT_ALREADY_EXISTS - Esiste già un contatto con questo msisdn
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

In caso il gruppo venga creato correttamente viene restituito lo stato HTTP status 200

Example:

```
curl --user AUTH_KEY:AUTH_SECRET -XPUT
https://api.smstools.it/rest/api/phonebook/contact/393480000000 \
-d "name=Mario" \
-d "lastname=Bianchi"
```

HTTP status 200

Elimina contatto

Eliminare un contatto in rubrica

URI

DELETE <https://api.smstools.it/rest/api/phonebook/contact/{msisdn}>

Request

parameter	Mandatory	Type	Description
msisdn	Yes (parameter in the URI)	String	Numero di cellulare del contatto compreso di prefisso internazionale

Response

HTTP Status Code

Code	Description
204	ok
401	Invalid credentials
404	Contact not found
405	HTTP Method not allowed
500	Generic Error

If the group is deleted correctly, the status “HTTP status 200” is returned

Example:

```
curl --user AUTH_KEY:AUTH_SECRET -XDELETE  
https://api.smstools.it/rest/api/phonebook/contact/393480000000
```

HTTP status 204

Custom field

Addressbook Custom field list

List of user-defined custom fields.

URI

GET <https://api.smstools.it/rest/api/phonebook/customfields/list>

Request

parameter	Mandatory	Type	Description
No parameter			

Response

HTTP Status Code

Code	Description
200	ok
401	Invalid credentials
405	HTTP Method not allowed

500	Generic Error
-----	---------------

The call returns a list with user-defined custom fields.

Each custom field is described by the following properties:

Property	Description
fieldKey	Unique key of the field
fieldName	Field Name
fieldType	Field Type Possible values are: TEXT NUMBER. DATE SELECT SELECTMULTIPLE

Example:

```
curl --user AUTH_KEY:AUTH_SECRET https://api.smstools.it/rest/api/phonebook/customfields/list
```

```
[
{
  "fieldKey": "f_indirizzo",
  "fieldName": "Indirizzo",
```

```
"fieldType": "TEXT"
},
{
  "fieldKey": "f_cap",
  "fieldName": "CAP",
  "fieldType": "NUMBER"
}
]
```

Resellers

- > New SubAccount
- > Edit SubAccount
- > Delete SubAccount
- > Upload SubAccount
- > Search SubAccount
- > Create Order
- > List of configured SMS packages
- > Remove credit SubAccount

In order to use the APIs listed on this page it is necessary to have aaccount **reseller** and request the specific **authorization**.

New SubAccount

Create a new SubAccount.

URI

POST <https://api.smstools.it/rest/api/reseller/subuser>

Request

The "Content-Type" for requests **POST** must be "**application / x-www-form-urlencoded**"

parameter	Mandatory	Type	Description
username	Yes	String	The username must be an alphanumeric string between 5 and 50 characters long. In addition to letters and numbers, it can contain the following characters .: @ _-
password	Yes	String	The password must be an alphanumeric string between 5 and 20 characters long.
name	Yes	String	Name
lastname	Yes	String	Surname
email	Yes	String	Valid email address
companyType	Yes	Boolean	Value to true if the account is associated with a company Possible values true / false

businessname	No (Yes if companyType is set to true)	Boolean	Company name
address	No	String	address
City	No	String	City
postcode	No	String	Postal code
province	No	String	Province
country	No	String	Country
birthdate	No	String	Date of birth in yyyy-MM-dd'T'HH:mm:ssZ format: mm:ssZ
phone	No	String	Phone
msisdn	No	String	Mobile
taxcode	No	String	Code Tax
vatnumber Number	No	String	VAT

Response

HTTP Status Code

Cod e	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - Bad account BAD_ACCOUNT_PERMISSION - Bad account Permission BAD_USERNAME - The username is not valid USERNAME_ALREADY_EXISTS - A user with this username already exists BAD_PASSWORD - The password is invalid BAD_NAME - parameter name not valid BAD_LASTNAME - parameter lastname not valid BAD_EMAIL - parameter email not valid BAD_BUSINESSNAME - parameter businessname not valid GENERIC_ERROR - internal server error
401	Invalid credentials

405	HTTP Method not allowed
500	Generic Error

UserData

Property	Description
id	Unique identifier of the SubAccount
name	Name
lastname	Surname
username	Username of the account
email	Address Email
businessname	Company name
address	address
city	City
Postcode	Zip code
provinces	Province
country	Country

birthdate	Date of birth
phonephone	PhoneMobile
msisdn	number
sender	Default Sender
senderAlias	List of senders use to send
taxcode	tax Code
vatnumber	VAT
registrationDate	Date in yyyy-mM-dd'T'HH:mm:ssZ
EXPIRATIONDATE	expiration date account in fomato yyyy-mM-dd'T ' HH: mm: ssZ Note: Your profile expires one year after your last order.
credit	credit
Remainingitalysms	Number of sms that can be sent to Italy
authKey	Auth Key API
authSecret	Secret Key API

Example:

```
curl --user AUTH_KEY: AUTH_SECRET -XPOST https://api.smstools.it/rest/api/reseller/subuser
-d "username = demoapirest"
-d "password = password"
-d "name = Mario "
-d "lastname = Rossi "
-d "email=info@apirest.cloud "
```

```
{
  "id ":" 1 ",
  "name ":" Mario ",
  "lastname ":" Rossi ",
  "username ":" demoapirest",
  "email": "info@apirest.cloud",
  "address": "",
  "city": "",
  "postcode": "",
  "province": "",
  "country": " ",
  "phone ":" ",
  "msisdn ":" ",
  "sender ":" smstools.it ",
  "registrationDate ":" 2015-08-04T12: 04: 56 + 0200 ",
  "expirationDate ":" 2016 -08-04T12: 04: 56 + 0200 ",
  "credit ": 0.6,
  "italysms ": 5,
  "italysmsLow ": 8,
  "senderAlias ": [],
  "authKey ":" 0000 ",
  "authSecret ":" 0000 "
}
```

Edit SubAccount

Edit a SubAccount.

URI

PUT [https://api.smstools.it/rest/api/reseller/subuser/ <id>](https://api.smstools.it/rest/api/reseller/subuser/<id>)

Request

The "**Content-Type**" for requests **PUT** must be "**application / x-www-form-urlencoded**"

parameter	Mandatory	Type	Description
name	Yes	String	Name
lastname	Yes	String	Surname
email	Yes	String	Valid email address
companyType	Yes	Boolean	Value to true if the account is associated with a company Possible values are true / false
businessname	No (Yes if companyType is set to true)	Boolean	Reason social company
address	No	String	address
City	No	String	City
postcode	No	String	Postal code
province	No	String	Province
country	No	String	Country

birthdate	No	String	Date of birth in yyyy-MM-dd'T'HH format: mm:ssZ
phone	No	String	Phone
msisdn	No	String	Mobile
taxcode	No	String	Code Tax
vatnumber Number	No	String	VAT

Response

HTTP Status Code

Cod e	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - The account is not a reseller BAD_ACCOUNT_PERMISSION - The account is not enabled to use this API SUBUSER_NOT_EXISTS - SubAccount not found BAD_NAME - parameter name not valid BAD_LASTNAME - parameter lastname not valid BAD_EMAIL - parameter email not valid BAD_BUSINESSNAME - parameter businessname not valid GENERIC_ERROR - internal server error
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

User Data

Refer to the call [new SubAccount](#) for more details

Example:

```
curl --user AUTH_KEY: AUTH_SECRET -XPUT https://api.smstools.it/rest/api/reseller/subuser/1
-d "name = Mario"
-d "lastname = Rossi"
```

```
-d "email=info@apirest.cloud "
```

```
{
  "id ":" 1 ",
  "name ":" Mario ",
  "lastname ":" Rossi ",
  "username ":" demoapirest ",
  "email ":" info @ apirest. cloud ",
  "address ":" ",
  "city ":" ",
  "postcode ":" ",
  "province ":" ",
  "country ":" ",
  "phone ":" ",
  "msisdn ":" ",
  "sender ":" smstools.it ",
  "registrationDate ":" 2015-08-04T12: 04: 56 + 0200 ",
  "expirationDate ":" 2016-08-04T12: 04: 56 + 0200 ",
  "credit ": 0.6,
  "italysms ": 5,
  "italysmsLow ": 8,
  "senderAlias ": [],
  "authKey ":" 0000 ",
  "authSecret ":" 0000 "
}
```

Delete SubAccount

Permanently deletes a SubAccount.

URI

DELETE [<id>](https://api.smstools.it/rest/api/reseller/subuser/)

Request

No parameter required

Response

HTTP Status Code

Cod e	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - Bad account type BAD_ACCOUNT_PERMISSION - Bad account permission SUBUSER_NOT_EXISTS - SubAccount not found
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Example:

```
curl --user AUTH_KEY: AUTH_SECRET -XDELETE https://api.smstools.it/rest/api/reseller/subuser/1
```

Load SubAccount

Load the full detail of a SubAccount.

URI

GET [https://api.smstools.it/rest/api/reseller/subuser/ <id>](https://api.smstools.it/rest/api/reseller/subuser/<id>)

Request

No parameter required

Response

HTTP Status Code

Cod e	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - BAD_ACCOUNT_PERMISSION - SUBUSER_NOT_EXISTS - SubAccount not found
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

User Data

Refer to the call [new SubAccount](#) for the details of the returned data

Example:

```
curl --user AUTH_KEY: AUTH_SECRET https://api.smstools.it/rest/api/reseller/subuser/1
```

```
{
  "id": "1",
  "name": "Mario",
  "lastname": " Rossi ",
  "username": " demoapirest ",
  "email": " info@apirest.cloud ",
  "address": " ",
  "city": " ",
  "postcode": " ",
  "province": " ",
  "country": "",
  "phone": "",
  "msisdn": "",
  "sender": "smstools.it",
  "registrationDate": "2015-08-04T12: 04: 56 + 0200",
  "expirationDate": "2016-08-04T12: 04: 56 + 0200",
  "credit": 0.6,
  "italysms": 5,
  "italysmsLow": 8,
  "senderAlias": [],
  "authKey": "0000",
  "authSecret": "0000"
}
```

Search SubAccount

Returns the list of SubAccounts

URI

GET <https://api.smstools.it/rest/api/reseller/subuser/search>

Request

parameter	Mandatory	Type	Description

query	No	String	SubAccount search by username, first name and last name
fromDate	No	String	Date from which the SubAccounts were registered in yyyy-MM-ddTHH format: mm:ssZ
toDate	No	String	Date on which the SubAccounts were registered in yyyy-MM-ddTHH format: mm:ssZ
offset	No	Integer	Position of the first result that will be returned. Example: If the search finds 100 sms, the offset field is set to 50 and the limit field to 10 will be returned the sms from 51 to 60. Default: 0
limit	No	Integer	Number of results Default: 20

Response

HTTP Status Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - BAD_ACCOUNT_PERMISSION - SUBUSER_NOT_EXISTS - SubAccount not found BAD_DATE - Format date not valid
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

User Data

Returns a list of SubAccounts. Refer to [new SubAccount](#) for more detail

Example:

```
curl --user AUTH_KEY: AUTH_SECRET https://api.smstools.it/rest/api/reseller/subuser/search
```

```
{
  "metadata": {
    "count": 1,
    "offset": 0,
    "limit": 20
  },
  "subUserList": [
    {
      "id": "1",
      "name": "Mario",
      "lastname": "Rossi",
```

```

"username": "demoapires",
"email": "info @ apirest. cloud ",
" address ":" ",
" city ":" ",
" postcode ":" ",
" province ":" ",
" country ":" ",
" phone ":" ",
" msisdn ":" ",
" sender ":" smstools.it ",
" registrationDate ":" 2015-08-04T12: 04: 56 + 0200 ",
" expirationDate ":" 2016-08-04T12: 04: 56 + 0200 ",
" credit ": 0.6,
" italysms ": 5,
" italysmsLow ": 8,
" senderAlias ": [],
" authKey ":" 0000 ",
" authSecret ":" 0000 "
}
]
}

```

Create Order

Create and activate an order for a SubAccount.

It is possible to assign an existing SMS package or alternatively create a new package by simply indicating the number of SMS assigned.

URI

POST <https://api.smstools.it/rest/api/reseller/subuser/<id>/order>

Request

The "**Content-Type**" for requests **POST** must be "**application / x-www-form-urlencoded**"

parameter	Mandatory	Type	Description

packId	No (Mandatory enhance at least one of the packId and newPackSmsItaly fields)	Integer	Identification of the SMS package to be activated. Use the call list packets to retrieve the identifiers
newPackSmsItaly	No (Mandatory value at least one of the field packId and newPackSmsItaly)	Integer	Number of SMS to assign by creating a new package
newPackUnitCostItaly	No	Double	Single cost used for a new package. It is used in combination with the parameter <i>newPackSmsItaly</i> Default: 0.07 €

Response

HTTP Code

Code	Description
200	ok

400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - BAD_ACCOUNT_PERMISSION - NO_PARAMS - No parameter set BAD_PACKID - parameter packId not valid BAD_SMS - parameter newPackSmsItaly not valid BAD_UNITCOST - parameter newPackUnitCostItaly not valid NO_CREDIT - the reseller does not have enough credit to complete the operation
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Example:

```
curl --user AUTH_KEY: AUTH_SECRET -XPOST https://api.smstools.it/rest/api/reseller/subuser/1/order
-d "newPackSmsItaly = 14"
```

List of SMS packages

Return the list of packages that can be ordered.

URI

GET <https://api.smstools.it/rest/api/reseller/pack/list>

Request

No data required

Response

HTTP Status Code

Cod e	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - BAD_ACCOUNT_PERMISSION -
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Data Package

Property	Description

id	package
smsItaly	number of SMS can be sent to Italy
unitCostItaly	single cost SMS italy
unitCostOtherZoneA	single cost SMS abroad area A
unitCostOtherZoneB	single cost SMS abroad zone B
unitCostOtherZoneC	single cost SMS abroad zone C
totalCost	total cost package
validFromDate	valid from date
validToDate	valid from date

Example:

```
curl --user AUTH_KEY: AUTH_SECRET https://api.smstools.it/rest/api/reseller/pack/list
```

```
[
{
  "id": 1,
  "smsItaly": 0,
  "unitCostItaly": 0.105,
  "unitCostOtherZoneA ":,
  0.084" unitCostOtherZoneB ": 0.105,
  " unitCostOtherZoneC ": 0.158,
  " totalCost ": 0,
  " validFromDate ":" 09.10.2004 ",
  " validToDate ":" 30.12.2005 "
}
```

Remove credit SubAccount

Remove credit from a SubAccount and reassign it to the reseller account.

URI

POST <https://api.smstools.it/rest/api/reseller/subuser/<id>/credit/remove>

Request

No parameter required

Response

HTTP Status Code

Cod e	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_ACCOUNT_TYPE - BAD_ACCOUNT_PERMISSION - SUBUSER_NOT_EXISTS - SubAccount not found
401	Invalid credentials

405	HTTP Method not allowed
500	Generic Error

Example:

```
curl --user AUTH_KEY: AUTH_SECRET -XPOST  
https://api.smstools.it/rest/api/reseller/subuser/1/credit/remove
```

Alias (SMS senders)

- > Alias List
- > New Alias
- > Clear Alias

List Alias

Returns the list of aliases.

Aliases are the **alphanumeric senders that can be** used for sending SMS.

URI

GET<https://api.smstools.it/rest/api/alias/list>

Request

No data requested

Response

HTTPStatus Code

Code	Description
200	OK
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Data Alias

Property	Description
id	ID of AliasAlias
alias	alias
vatnumber	VAT number
taxcode	Code tax
businessname	company name
address	address
city	city

postcode	CAP
province	province
country	Country
email	email
phone	phone
status	State. The possible states are: VERIFIED - alias verified NOTVERIFIED - alias not usable

Example:

```
curl --user AUTH_KEY: AUTH_SECRET https://api.smstools.it/rest/api/alias/list
```

```
[
  {
    "id": 13497,
    "alias": "test",
    "vatnumber": "01000000000",
    "taxcode": "01000000000",
    "businessname": "API Rest",
    "address": "via Rossi, 21",
    "city": "milano",
```



```
"postcode": "20143",  
"provinces": "mi",  
"country": "IT",  
"email": "info@apirest.cloud",  
"phone": "021230321",  
"status": "VERIFIED"  
}  
]
```

New Alias

Create a new Alias

URI

POST <https://api.smstools.it/rest/api/alias>

Request

The "**Content-Type**" for requests **POST** must be "**application / x-www-form-urlencoded**"

parameter	Mandatory	Type	Description
Alias	Yes	String	identification string. The string cannot exceed 11 characters. It can contain only uppercase or lowercase letters (az), numbers, spaces and characters. _ - & @ * "

businessname	Yes	String	company name to associate with the alias
address	Yes	String	address of the company to be associated with the alias
city	Yes	string	city
postcode	Yes	String	ZIP code of the company headquarters to associate with the alias
province	Yes	String	province of the company headquarters to associate with the alias
country	Yes	String	nation of the company headquarters to associate with the alias. Specify the ISO 3166-1 two-letter code. Ad Example for Italy specify IT
vatnumber	Yes	String	VAT number of the company to be associated with the alias
email	Yes	String	email of the company to associate with the alias
phone	Yes	String	telephone of the company to be associated with the alias

taxcode	Yes	String	Code tax company to associate to the alias
---------	-----	--------	--

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: BAD_ALIAS - Alias absent or not valid BAD_BUSINESSNAME - parameter businessname not valid BAD_VATNUMBER - parameter vatnumber not valid BAD_ADDRESS - parameter address not valid BAD_CITY - parameter city not valid BAD_POSTCODE - parameter postcode not valid BAD_PROVINCE - parameter province not valid BAD_COUNTRY - parameter country not valid

	BAD_EMAIL - parameter email not valid BAD_PHONE - parameter phone not valid BAD_TAXCODE - parameter taxcode not valid GENERIC_ERROR - internal server error
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Data Alias

In case of success the data of the newly created Alias will be returned

Example:

```
curl --user AUTH_KEY: AUTH_SECRET -XPOST https://api.smstools.it/rest/api/alias
-d "alias = test"
-d "vatnumber = 01000000000"
-d "taxcode = 01000000000" \
-d "BUSINESSNAME Rest API =" \
-d "address = via Rossi, 21" \

-d "milano city =" \

-d "postcode = 20143" \
```

```
-d "province = I" \  
  
-d "BUSINESSNAME = info@apirest.cloud "  
  
d" email = IT "  
  
d" phone = 021230321 "  
  
{  
  " id ": 13497,  
  
  " alias ":" test ",  
  
  " vatnumber ":" 01000000000 "  
  
  " taxcode ":" 01000000000 "  
  
  " businessname ":" API Rest ",  
  
  " address ":" via Rossi, 21 ",  
  
  " city ":" milano ",  
  
  " postcode ":" 20143 ",  
  
  " province ":" mi ",  
  
  " country " : "IT",  
  
  "email": "info@apirest.cloud",  
  
  "phone": "021230321",  
  
  "status": "VERIFIED"Permanently deletes  
}
```

Cancel Alias

Cancel an Alias.

URI

DELETE [https://api.smstools.it/rest/api/alias/ <id>](https://api.smstools.it/rest/api/alias/<id>)

Request

No parameter required

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: ALIAS_NOT_EXISTS - Alias not found
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Example:

```
curl --user AUTH_KEY: AUTH_SECRET -XDELETE https://api.smstools.it/rest/api/alias/13497
```

SMS OTP

- > Enter Code
- > Check Code
- > Cancel Code
- > Search Code

Submit Code

Send a numeric code to a recipient.

You can test the submission by adding the parameter `sandbox` to true in the Request.

URI

POST <https://api.smstools.it/rest/api/verify/send>

Request

The "Content-Type" for requests **POST** must be "**application / x-www-form-urlencoded**"

parameter	Mandatory	Type	Description
to	Yes	String	The recipient of the message. It must be a mobile number including an international prefix. Ad Example: 393481234567
text	Yes	String	Message Text If the message exceeds 160 characters, multiple concatenated SMS messages are sent up to a maximum of 5. The text may contain a placeholder which will be replaced during submission by the Code. The placeholder must be specified with the syntax \$ {verify_code}

			Ad Example: text = Rest Code API \$ {verify_code}
from	No	String	The sender of the SMS. It can be a validated telephone number or an alphanumeric string of 11 characters. If not specified, the default sender of his profile is used.
sandbox	No	Boolean	Allows you to test the call without actually sending the message. useful parameter in the development phase of possible integrations with the REST API. Possible values true / false
app_id	No	String	Identifies the application for which the Code is requested. You can specify your own identifier Warning: You cannot generate 2 codes active simultaneously for the same recipient for the same application

encoding	No	String	The encoding from use for the Message Text Possible values are: 7BIT - Standard encoding for GSM messages. The maximum length for a single text message is 160 characters. UCS2 - 8-bit encoding to send GSM messages with non-ASCII content. The maximum length for a single text message is 70 characters. AUTO - The encoding to be used is automatically identified based on the content of the text field. Default: 7BIT
code_len	No	Integer	Code Length Possible values are from 4 to 6. Default: 4
max_retry	No	Integer	Maximum number of attempts allowed. The possible values are from 1 to 20. Default: 3

ttl	No	Integer	maximum duration in seconds of the Code. The minimum value is 10. Default: 300
-----	----	---------	--

Response

HTTPStatus Code

Code	Description
200	OK
400	Parameter not valid Possible errorMsg are: SPAM - were sent more than 10 text messages in the last 60 minutes at the same destination. BAD_TO - parameter to not valid BAD_TEXT_PLACEHOLDER - text missing \$ {verify_code} or contains placeholders irreplaceable BAD_TEXT - parameter text not valid BAD_FROM - parameter from not valid BAD_ENCODING - parameter encoding not valid

	BAD_APP_ID - parameter app_id not valid BAD_COD_LEN - parameter code_len not valid BAD_RETRY - parameter max_retry not valid BAD_TTL - parameter ttl not valid NO_CREDIT - No Credit GENERIC_ERROR - internal server error
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Response

Property	Description
verify_id	Unique ID of Request Code.

Example:

CURL
JAVA
PHP
PYTHON

C

```
curl --user AUTH_KEY: AUTH_SECRET  
-XPOST https://api.smstools.it/rest/api/verify/send  
-d "to = 393480000000"  
-d "text = Code of unlock # {verify_code} "
```

```
{  
  "verify_id ":" c6cf59524c65d9c65d134d1137a86336558d19ba "  
}
```

Check Code

Check if the Code matches the one sent by SMS.

URI

GET <https://api.smstools.it/rest/api/verify/check>

Request

parameter	Mandatory	Type	Description
verify_id	Yes	String	The Code received from the send command.
verify_code	Yes	String	The number to test.
ip_address	No	String	Ip of the user to allow to implement an antispam filter.

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: VERIFY ID ERROR - parameter verify_id not valid
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Response

Property	Description
id_sms	Id of SMS sent. Yes can use with https://api.smstools.it/rest/api/verify/search to retrieve the message
verify_status VERIFIED	verification result - The Code corresponds to FAILED - The Code does NOT match EXPIRED - The Code has expired

	VERIFY ID ERROR - The parameter verify_id is incorrect. REUSED - The Code has already been verified. CANCELLED - The code has been canceled
--	---

Example:

CURL

JAVA

PHP

PYTHON

C #

```
curl --user AUTH_KEY: AUTH_SECRET \
```

```
-XGET
```

```
"https://api.smstools.it/rest/api/verify/check?verify_id=728d03eee5a72e8102a4a905e6325449d27c583b&verify_code=1234"
```

```
{
```

```
  "id_sms": "410065822",
```

```
  "verify_status": "VERIFIED"
```

```
}
```

Cancel Code

Cancels an active Code.

It allows to cancel a still usable Code.

URI

POST <https://api.smstools.it/rest/api/verify/command>

Request

The "Content-Type" for requests **POST** must be "application / x-www-form-urlencoded"

parameter	Mandatory	Type	Description
verify_id	Yes	String	The code received from the send command.
command	Yes	String	Can take the value: DELETE

Response

HTTP Code Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: VERIFY ID ERROR - parameter verify_id is invalid BAD STATUS - the Request is not erasable, it is already unusable BAD_COMMAND - parameter command is invalid
401	Invalid credentials
405	HTTP Method not allowed

500	Generic Error
-----	---------------

Response

Property	Description
command	Command executed

Example:

CURL

JAVA

PHP

PYTHON

C #

```
curl --user AUTH_KEY: AUTH_SECRET -XPOST https://api.smstools.it/rest/api/verify/command \
-d "verify_id 35baca938c120e49e462809b4e62efb5fe6f6bd8 =" \
-d "command DELETE ="
```

```
{
  "command": "DELETE"
}
```

Search Code

Retrieves the history of a Code.

URI

GET <https://api.smstools.it/rest/api/verify/search>

Request

parameter	Mandatory	Type	Description
-----------	-----------	------	-------------

verify_id	Yes	String	The Code received from the send command .
-----------	-----	--------	---

Response

HTTP Status Code

Code	Description
200	ok
400	Parameter not valid Possible errorMsg are: NO_PARAMS - No parameter set VERIFY ID ERROR - parameter verify_id not valid
401	Invalid credentials
405	HTTP Method not allowed
500	Generic Error

Response

Property	Description
verify_id	request id

verify_status	request status
to	destination of the SMS
from the	sender of the message
id_sms	id of the sent SMS.
Code	Code sent. Populated only if sandbox = true.
date_submit	date the message is sent.
valid_until	end of validity date.
checks	enlistof registered checks. date - data check verify_code - sent code verify_status - outcome ip_address -ip_address sent

Example:

CURL

JAVA

PHP

PYTHON

C #

curl --user AUTH_KEY: AUTH_SECRET \

-XGET "https://api.smstools.it/rest/api/verify/search?verify_id=728d03eee5a72e8102a4a905e6325449d27c583b"

```
{
  "verify_id": "745ab0c7d2bebd93e7e114ec973a1c997182fcbf",
  "verify_status": "VERIFIED ",
  "to": "393480000000 ",
  "from": "393480000000 ",
  "id_sms": "410065825 ",
  "date_submit": 1480499821000,
  "valid_until": 1480500121000,
  "checks": [{
    "date": 1480499858000,
    "verify_code": "4506 ",
    "verify_status": "VERIFIED ",
    "ip_address": "192.168.1.1 "
  }]
}
```