

Table of contents

1. Webservice.....	3
2. Structure.....	3
3. Authentication	3
Legacy auth	4
4. Lists.....	4
GET /lists.{format}.....	4
GET /list/{id}.{format}	4
POST /list.{format}	5
PUT /list/{id}.{format}	6
DELETE /list/{id}.{format}.....	7
5. Contacts.....	7
GET /contacts/{page}.{format}	7
GET /contact/{id}.{format}.....	9
POST /contact.{format}.....	9
PUT /contact/{id}.{format}.....	10
DELETE /contact/{id}.{format}	12
6. Contact import	12
POST /import.{format}.....	12
PUT /import/{hash}.{format}	13
GET /import/{hash}.{format}	14
7. Merge fields	15
GET /mergefields.{format}.....	15
POST /mergefield.{format}	15
PUT /mergefield/{id}.{format}	16
DELETE /mergefield/{id}.{format}.....	16
8. Messages.....	17
POST /message.{format}.....	17
Format.....	17
UNICODE	18
BINARY.....	19
WAPPUSH	19
MMS	20
Charged messages.....	20

GET /messages/{page}.{format}	25
GET /message/{batchid}/{page}.{format}.....	26
PUT /message/refund/{lookup-id}.{format}.....	27
9. Query interface	28
GET /inbox/{page}.{format}	28
GET /inbox/lookup/{lookupid}.{format}.....	29
GET /outbox/{page}.{format}.....	29
GET /outbox/lookup/{lookupid}.{format}.....	31
10. Stats.....	32
GET /stats/{type}/{aggregate}.{format}.....	32
POST /stats/inbox.{format}.....	35
POST /stats/outbox.{format}	36



1. Webservice

The webservice is at <http://api.zumosms.com> and <https://api.zumosms.com>

API KEY is found at <https://sms.zumosms.com> under Customerinfo (Indstillinger)

2. Structure

All api calls has a version in the URL. We will only bump the version in the api call in any new features that would break backward compatibility are added.

We support json and xml

An url will look something like this

```
http://api.zumosms.com/\[version\]/\[resource\].\[json|xml\]?apikey=\[apikey\]  
curl 'http://api.zumosms.com/v2/ping.json?apikey=  
e1bd9f55b7d6a45348fa92c427932c4ad3744afdeb8b99aa207de991f78f9d4' -X GET  
curl 'http://api.zumosms.com/v2/ping.xml?apikey=  
e1bd9f55b7d6a45348fa92c427932c4ad3744afdeb8b99aa207de991f78f9d4' -X GET
```

3. Authentication

Authentication is done by sending the param `apikey=[apikey]`

```
curl 'http://api.zumosms.com/v2/ping.json?apikey=[apikey]' -X GET  
curl 'http://api.zumosms.com/v2/ping.xml?apikey=[apikey]' -X GET
```

and you'll get something like:

```
<?xml version="1.0"?>  
<response>  
  <status>200</status>  
  <userid>7405</userid>  
  <username>hn</username>  
  <iprange>77.243.40.100</iprange>  
  <apikey>a1bd9f58b7d6a45b48fa92c427132c4ad3744afdeb8b99af207de991f78f9d4</apikey>  
</response>  
{  
  "status": 200,  
  "userid": "7405",  
  "username": "hn",  
  "iprange": "77.243.40.100",  
  "apikey": "a1bd9f58b7d6a45b48fa92c427132c4ad3744afdeb8b99af207de991f78f9d4"  
}
```

Legacy auth

This function should not be used in new implementations, and will in the future be deprecated.

```
curl 'http://api.zumosms.com/v2/ping.json?apikey=legacy&username=[username]&password=[password]&type=[md5|plain]' -X GET
curl 'http://api.zumosms.com/v2/ping.xml?apikey=legacy&username=[username]&password=[password]&type=[md5|plain]' -X GET
```

4. Lists

GET /lists.{format}

Get all lists

```
curl 'http://api.zumosms.com/v2/lists.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/lists.xml?apikey=[apikey]' -X GET
{
  "list": [
    {
      "id": "1",
      "parent": "0",
      "name": "HMN"
    },
    {
      "id": "9",
      "parent": "0",
      "name": "test42"
    }
  ],
  "status": 200
}
```

GET /list/{id}.{format}

Get specific list and all the contacts in it

```
curl 'http://api.zumosms.com/v2/list/{id}.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/list/{id}.xml?apikey=[apikey]' -X GET
{
  "list": {
    "id": "33",
    "parent": "9",
  }
}
```

```

"name": "Underliste 2",
"contacts": [
  {
    "id": "85",
    "listid": "9",
    "mobile": "+4587654321",
    "created": "2014-12-15 13:51:26",
    "changed": "2014-12-15 13:51:26",
    "fields": {
      "Fornavn": "Jack",
      "Efternavn": "Bauer"
    }
  },
  {
    "id": "93",
    "listid": "9",
    "mobile": "+4512345678",
    "created": "2014-12-15 13:52:32",
    "changed": "2014-12-15 13:52:32",
    "fields": {
      "Fornavn": "Anders",
      "Efternavn": "And"
    }
  }
]
},
"status": 200
}

```

POST /list.{format}

Create new list

```

curl 'http://api.zumosms.com/v2/list.json?apikey=[apikey]' -X POST -H
"Content-Type: application/json" -d @file.json
curl 'http://api.zumosms.com/v2/list.xml?apikey=[apikey]' -X POST -H
"Content-Type: text/xml" -d @file.xml

```

payload

file.json

```

{
  "list":{
    "parent": 0,
    "name":"test123"
  }
}

```

file.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<list>
  <parent>0</parent>
  <name>test42</name>
</list>
```

response

```
{
  "list": {
    "id": "61",
    "parent": "0",
    "name": "test123",
    "contacts": []
  },
  "status": 201
}
<?xml version="1.0"?>
<response>
  <list>
    <id>69</id>
    <parent>0</parent>
    <name>test42</name>
    <contacts/>
  </list>
  <status>201</status>
</response>
```

PUT /list/{id}.{format}

Update existing list

```
curl 'http://api.zumosms.com/v2/list/{id}.json?apikey=[apikey]' -X PUT -H
"Content-Type: application/json" -d @file.json
curl 'http://api.zumosms.com/v2/list/{id}.xml?apikey=[apikey]' -X PUT -H
"Content-Type: text/xml" -d @file.xml
```

payload

file.json

```
{
  "list":{
    "parent": 0,
    "name":"test123"
  }
}
```

file.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<list>
  <parent>0</parent>
  <name>test42</name>
</list>
```

response

```
{
  "list": {
    "id": "61",
    "parent": "0",
    "name": "test123",
    "contacts": []
  },
  "status": 201
}
<?xml version="1.0"?>
<response>
  <list>
    <id>69</id>
    <parent>0</parent>
    <name>test42</name>
    <contacts/>
  </list>
  <status>201</status>
</response>
```

DELETE /list/{id}.{format}

Delete a specific list

```
curl 'http://api.zumosms.com/v2/list/{id}.json?apikey=[apikey]' -X DELETE
curl 'http://api.zumosms.com/v2/list/{id}.xml?apikey=[apikey]' -X DELETE
{
  "message": "List id [65] deleted",
  "status": 200
}
```

5. Contacts

GET /contacts/{page}.{format}

Get all contacts based on search params

```
curl 'http://api.zumosms.com/v2/contacts/1.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/contacts/1.xml?apikey=[apikey]' -X GET
```

Fields

You can use the following GET parameter when searching for contacts

Name	Type	Description
listid	String	Search for contacts in a specific list
mobile	String	Search for contacts with a specific mobile number
deleted	Boolean	Search for contacts that are deleted or active
field_key	String	Search merge field key
field_value	String	Search merge field value

response

```
{
  "contacts": [
    {
      "id": "631809",
      "listid": "373",
      "mobile": "+4512345678",
      "created": "2016-02-13 09:11:22",
      "changed": "2016-02-13 09:11:22",
      "deleted": false,
      "fields": {
        "Fornavn": "TEST",
        "Efternavn": ""
      }
    },
    {
      "id": "632065",
      "listid": "373",
      "mobile": "+4512345678",
      "created": "2016-02-13 09:11:22",
      "changed": "2016-02-13 09:11:22",
      "deleted": false,
      "fields": {
        "Fornavn": "TEST",
        "Efternavn": ""
      }
    }
  ],
  "paging": {
    "pages": 81056,
    "offset": 2,
    "current": 1
  }
}
```



```
    },
    "status": 200
  }
}
```

GET /contact/{id}.{format}

Get one contact by id

```
curl 'http://api.zumosms.com/v2/contact/{id}.json?apikey=[apikey]' -X GET
```

```
curl 'http://api.zumosms.com/v2/contact/{id}.xml?apikey=[apikey]' -X GET
```

```
{
  "contact": {
    "id": "85",
    "listid": "9",
    "mobile": "+4512345678",
    "created": "2014-12-15 13:51:26",
    "changed": "2014-12-15 13:51:26",
    "fields": {
      "Fornavn": "Jack",
      "Efternavn": "Bauer"
    }
  },
  "status": 200
}
```

POST /contact.{format}

Create new contact

```
curl 'http://api.zumosms.com/v2/contact.json?apikey=[apikey]' -X POST -H
```

```
"Content-Type: application/json" -d @file.json
```

```
curl 'http://api.zumosms.com/v2/contact.xml?apikey=[apikey]' -X POST -H
```

```
"Content-Type: text/xml" -d @file.xml
```

payload

file.json

```
{
  "contact": {
    "listid": "33",
    "mobile": "+4512345678",
    "fields": {
      "Fornavn": "Jack",
      "Efternavn": "Bauer"
    }
  }
}
```

file.xml

```
<?xml version="1.0"?>
<contact>
  <listid>33</listid>
  <mobile>+4512345678</mobile>
  <fields>
    <Fornavn>Jack</Fornavn>
    <Efternavn>Bauer</Efternavn>
  </fields>
</contact>
```

response

```
{
  "contact": {
    "id": "109",
    "listid": "33",
    "mobile": "+4512345678",
    "created": "2014-12-17 08:32:31",
    "changed": "2014-12-17 08:32:31",
    "fields": {
      "Fornavn": "Jack",
      "Efternavn": "Bauer"
    }
  },
  "status": 201
}
```

```
<?xml version="1.0"?>
<response>
  <contact>
    <id>113</id>
    <listid>33</listid>
    <mobile>+4512345678</mobile>
    <created>2014-12-17 08:32:51</created>
    <changed>2014-12-17 08:32:51</changed>
    <fields>
      <Fornavn>Jack</Fornavn>
      <Efternavn>Bauer</Efternavn>
    </fields>
  </contact>
  <status>201</status>
</response>
```

PUT /contact/{id}.{format}

Update existing contact

```
curl 'http://api.zumosms.com/v2/contact/{id}.json?apikey=[apikey]' -X PUT -H
"Content-Type: application/json" -d @file.json
```

```
curl 'http://api.zumosms.com/v2/contact/{id}.xml?apikey=[apikey]' -X PUT -H  
"Content-Type: text/xml" -d @file.xml
```

payload

file.json

```
{  
  "contact": {  
    "listid": "33",  
    "mobile": "+4512345678",  
    "fields": {  
      "Fornavn": "Anders",  
      "Efternavn": "And"  
    }  
  }  
}
```

file.xml

```
<?xml version="1.0"?>  
<contact>  
  <listid>33</listid>  
  <mobile>+4512345678</mobile>  
  <fields>  
    <Fornavn>Anders</Fornavn>  
    <Efternavn>And</Efternavn>  
  </fields>  
</contact>
```

response

```
{  
  "contact": {  
    "id": "109",  
    "listid": "33",  
    "mobile": "+4512345678",  
    "created": "2014-12-17 08:32:31",  
    "changed": "2014-12-17 08:32:31",  
    "fields": {  
      "Fornavn": "Anders",  
      "Efternavn": "And"  
    }  
  },  
  "status": 201  
}  
<?xml version="1.0"?>  
<response>  
  <contact>  
    <id>113</id>
```

```
<listid>33</listid>
<mobile>+4512345678</mobile>
<created>2014-12-17 08:32:51</created>
<changed>2014-12-17 08:32:51</changed>
<fields>
  <Fornavn>Anders</Fornavn>
  <Efternavn>And</Efternavn>
</fields>
</contact>
<status>201</status>
</response>
```

DELETE /contact/{id}.{format}

Delete a specific contact by id

```
curl 'http://api.zumosms.com/v2/contact/{id}.json?apikey=[apikey]' -X DELETE
curl 'http://api.zumosms.com/v2/contact/{id}.xml?apikey=[apikey]' -X DELETE
{
  "message": "Contact id [113] deleted",
  "status": 200
}
{
  "status": 404,
  "message": "Contact id [113] not found"
}
```

6. Contact import

POST /import.{format}

Initialize import job

```
curl 'http://api.zumosms.com/v2/import.json?apikey=[apikey]' -X POST -H
"Content-Type: application/json" -d @file.json
curl 'http://api.zumosms.com/v2/import.xml?apikey=[apikey]' -X POST -H
"Content-Type: text/xml" -d @file.xml
```

Fields

Name	Type	Description
listid	integer	If you want to import into an existing list, add the listid
name	String	Name to use for new list, only used if no listid is set
clean	Boolean (1 = delete, 0 = keep)	Should all contacts in the list be deleted before import

Name	Type	Description
duplicate	Boolean (1 = delete, 0 = keep)	Should duplicate numbers be deleted

payload

file.json

```
{
  "import": {
    "listid": 33,
    "name": "TEST",
    "clean": 0,
    "duplicate": 0
  }
}
```

response

```
{
  "import": {
    "hash": "1fa97e90d173a9ed979bec56f3cef0f1d5c2f3fbf33e263dcbae28c9bd006abe",
    "created": "2014-12-17 15:24:57",
    "listid": 0,
    "name": "TEST",
    "clean": 0,
    "total": 0,
    "processed": 0
  },
  "status": 201
}
```

PUT /import/{hash}.{format}

List of numbers to import

Uses {hash} from POST

```
curl 'http://api.zumosms.com/v2/import/{hash}.json?apikey=[apikey]' -X PUT -H
"Content-Type: application/json" -d @file.json
curl 'http://api.zumosms.com/v2/import/{hash}.xml?apikey=[apikey]' -X PUT -H
"Content-Type: text/xml" -d @file.xml
```

payload

file.json

```
{
  "import": {
    "keys": [
```

```

        "Mobile",
        "Fornavn",
        "Efternavn"
    ],
    "values": [
        [
            "+4512345678",
            "Jack",
            "Bauer"
        ],
        [
            "+4587654321",
            "Chuck",
            "Norris"
        ]
    ]
}
}

```

response

```

{
  "import": {
    "hash": "792d979f57f8f046ba3c5d73353be07d4fa06a9cfbe491f044be1d357f63756c",
    "created": "2014-12-17 15:54:01",
    "listid": 1,
    "name": "TEST",
    "clean": 0,
    "duplicate": 1,
    "total": 3,
    "processed": 3
  },
  "errors": [
    {
      "message": "Mobile is not valid! Remember a valid number must contain a country",
      "count": 1,
      "errors": {
        "1": "4587654321"
      }
    }
  ],
  "status": 200
}

```

GET /import/{hash}.{format}

Get status on import

```
curl 'http://api.zumosms.com/v2/import/{hash}.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/import/{hash}.xml?apikey=[apikey]' -X GET
{
  "import": {
    "hash": "1fa97e90d173a9ed979bec56f3cef0f1d5c2f3fbf33e263dcbae28c9bd006abe",
    "created": "2014-12-17 15:24:57",
    "listid": 0,
    "name": "TEST",
    "clean": 0,
    "total": 100,
    "processed": 100
  },
  "status": 200
}
```

7. Merge fields

GET /mergefields.{format}

Get all merge fields

```
curl 'http://api.zumosms.com/v2/mergefields.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/mergefields.xml?apikey=[apikey]' -X GET
{
  "field": [
    {
      "id": "1",
      "name": "Fornavn",
      "key": "felt1"
    },
    {
      "id": "5",
      "name": "Efternavn",
      "key": "felt5"
    }
  ],
  "status": 200
}
```

POST /mergefield.{format}

Create new merge field

```
curl 'http://api.zumosms.com/v2/mergefield.json?apikey=[apikey]' -X POST -H
"Content-Type: application/json" -d @file.json
curl 'http://api.zumosms.com/v2/mergefield.xml?apikey=[apikey]' -X POST -H
"Content-Type: text/xml" -d @file.xml
```

payload

file.json

```
{
  "field":{
    "name":"Test"
  }
}
```

response

```
{
  "field": {
    "id": "77",
    "name": "Test",
    "key": "felt77"
  },
  "status": 201
}
```

PUT /mergefield/{id}.{format}

Update existing merge field

```
curl 'http://api.zumosms.com/v2/mergefield/{id}.json?apikey=[apikey]' -X PUT -H
"Content-Type: application/json" -d @file.json
curl 'http://api.zumosms.com/v2/mergefield/{id}.xml?apikey=[apikey]' -X PUT -H
"Content-Type: text/xml" -d @file.xml
```

payload

file.json

```
{
  "field":{
    "name":"Test changed"
  }
}
```

response

```
{
  "field": {
    "id": "73",
    "name": "Test changed",
    "key": "felt73"
  },
  "status": 200
}
```

DELETE /mergefield/{id}.{format}

Delete a specific contact by id

```
curl 'http://api.zumosms.com/v2/mergefield/{id}.json?apikey=[apikey]' -X DELETE
curl 'http://api.zumosms.com/v2/mergefield/{id}.xml?apikey=[apikey]' -X DELETE
{
  "message": "Merge field id [113] deleted",
  "status": 200
}
{
  "status": 404,
  "message": "Merge field id [113] not found"
}
```

8. Messages

POST /message.{format}

Here are some examples:

Format

GSM

Sending a normal message

A simple example

```
{
  "message":{
    "recipients": "+4512721272",
    "sender": "TEST",
    "message": "Hello world!"
  }
}
```

A simple example with a number as sender

```
{
  "message":{
    "recipients": "+4512721272",
    "sender": "+4512345678",
    "message": "Hello world!"
  }
}
```

An more advanced example using a lot of the available parameters

```
{
  "message":{
    "format": "GSM",
    "recipients": "1,+4512721272",
    "sender": "TEST",
    "message": "Hello world!",
    "status": true,
    "statusurl": "http://example.com/dlr.php",
    "returndata": "my-own-data",
    "class": 0
  }
}
```

Send a message to a list

```
{
  "message":{
    "recipients": "1",
    "sender": "Test",
    "message": "Hello world"
  }
}
```

Send a time scheduled message to a list.

The message will be sent at 30-05-2015 10:00

```
{
  "message":{
    "recipients": "1",
    "sender": "Test",
    "message": "Hello world",
    "sendtime": "30-05-2015 10:00"
  }
}
```

Send a message to more recipients

```
{
  "message":{
    "recipients": "1,+4512345678,+4522112497,3",
    "sender": "Test",
    "message": "Hello world"
  }
}
```

UNICODE

To send speciality chars like chinese letters.

A normal message is 160 chars, but if you use unicode each message can only hold 70 chars

Send the text "ÛÑÎÇÔÐÊ"

```
{
  "message":{
    "recipients": "+4512345678",
    "sender": "TEST",
    "message": "ÛÑÎÇÔÐÊ",
    "format": "UNICODE",
    "charset": "UTF-8"
  }
}
```

BINARY

With binary messages you can send special message like a nokia vCard or similar.

To send a binary message you need to define the body in hex and set udh

Example of a Bookmark

```
{
  "message":{
    "recipients": "+4512345678",
    "sender": "TEST",
    "message": "01062D1F2B6170706C69636174696F6E2F782D7761702D70726F762E62726F777365722D626F",
    "format": "BINARY",
    "udh": "0C05040B8423F00504C34FC002"
  }
}
```

WAPPUSH

Send a link that is opened on the phone

You need to define pushurl as well as the other required parameters for WAPPUSH messages

Example of a link to jp.dk

```
{
  "message":{
    "recipients": "+4512345678",
    "sender": "TEST",
    "message": "Jyllands Posten",
    "format": "WAPPUSH",
    "pushurl": "http://jp.dk"
  }
}
```

MMS

To send a MMS you need to send an Array of attachments.

To send a presentation, the first attachment needs to be a SMIL document with the extension .smil
Sender should be a valid shortcode

Example of a simple MMS message with one image

```
{
  "message":{
    "recipients": "+4512345678",
    "sender": "1272",
    "message": "MMS test",
    "format": "MMS",
    "charset": "UTF-8",
    "attachment": [
      "http://yourdomain.com/sms.png"
    ]
  }
}
```

Charged messages

If you have access to send charged messages you need to define a price and contenttype

Send a message with a price

Price is 10dkk(1000) to a single number (NOTE: sender has to be a valid shortcode)

```
{
  "message":{
    "recipients": "+4512345678",
    "sender": "1272",
    "message": "Hello world",
    "price": 1000,
    "contenttype": 9,
    "invoicetext": ""
  }
}
```

Send message.

```
curl 'http://api.zumosms.com/v2/message.json?apikey=[apikey]' -X POST -H
"Content-Type: application/json" -d @file.json
```

```
curl 'http://api.zumosms.com/v2/message.xml?apikey=[apikey]' -X POST -H
"Content-Type: text/xml" -d @file.xml
```

Fields

Name	Type	Default	Description
recipients	String		Comma separated list of recipients. listid, contactid or phone number, see format below
sender	String(11) / Int(22)	Service SMS	Name / Number of sender. If number defined use this format: +4512345678 where +45 is countrycode
message	String		The message to send
status	Boolean	false	Should a status request be made. See [api:example_dlr]
statusurl	String		URL to request with status of each message. See [api:example_dlr]
returndata	String		Custom data to return in statusurl request. See [api:example_dlr]
class	Int	1	See class list below
sendtime	Date		Send message at a later time, Format : dd-mm-YYYY HH:MM
price	Int	0	Price in local unit for the selected shortcode. For 1272 in DK it would be øre
charity	Boolean	false	Set price to be for a charity. Only allowed for approved users.
invoicetext	String		Invoice text that should be shown on end users phone bill
validity	Int	2880	Message TTL in minutes
contenttype			
format	Enum	GSM	See list below
udh	String		Set UDH, used by BINARY messages
attachment	Array		Used by format MMS
pushurl	String		Used by format WAPPUSH. The url pushed to the terminal

Name	Type	Default	Description
pushexpire	String		Used by format WAPPUSH. Exire time in unix timestamp, default is now + 7 days
filter	Array		Filter who in a list to send the message to. Requires the use of lists and mergefields.
segmentation	Array		Segmentation options for splitting lists into smaller chunks dynamicly when sending.
pid	Int	0	GSM Protocol ID
advanced	String		Key for using advanced fields
protocol	String (advanced)		The protocol name (REST, SMTP and others)

recipients

Name	Format	Description
phone number	+ [number]	Phone numbers prefix is +. For instance +4512345678
contact id	c[id]	Contact id prefix is c. For instance c123
list id	[id]	List id is default and only an integer

recipients could look like this

1, 2, c123, +452721272

format

Name	Required	Description
GSM		Send normal message
UNICODE		To send speciality chars like chinese letters. A normal message is 160 chars, but if you use unicode each message can only hold 70 chars
BINARY	udh, message	Send a binary message body in hex and define udh
WAPPUSH	pushurl, pushexpire	Send a link that is opened on the phone
MMS	attachment	Array of attachments to send as MMS To send a presentation, the first attachment needs to be a SMIL document with the extension .smil Sender should be a valid shortcode

Keyword replacement

If you are sending to lists, you can use keyword replacement, for instance for inserting a unique name in each message.

If your list had an extra field called Name you would use it like this :

Hi [Name]

This is only a test

contenttypes

- 1 = Ringetoner og billeder
- 2 = Videoklip og tv
- 3 = Voksenindhold
- 4 = Musik
- 5 = Lydbøger og podcasts
- 6 = Mobilspil
- 7 = Chat tjenester
- 8 = Konkurrence og afstemning
- 9 = M-payment (Fysik varer)
- 10 = Nyheder og information
- 11 = Donationer (Humanitær)
- 12 = Telemetri (M2M)
- 13 = Diverse
- 14 = Donationer (Forening)

class

- 0 - Show message directly on phone. The message is not saved on the phone. (Also known as flash messages)
- 1 - Save message in phone memory. Either on the phone or in SIM.
- 2 - Message contains SIM data.
- 3 - Message contains info that indicate that it should be sent to external units, normally used by terminal equipment.

By default you should not change this option

filter

Filter is a json array of rules that is matched during the send process, to limit who should get the message.

Name	Values	Description
match	any / all	The way the rules should be matched. Should all or any of the rules match to include a recipient.

Name	Values	Description
rules	field, operation, value	Array of rules to match on
Operation	value	Descriptuon
is	is	If exact match is found include the recipient
is not	not	If exact is not found include the recipient
contains	contains	If mergefield contains the string include the recipient
does not contain	notcontain	If mergefield does not contain the string then include the recipient
starts with	start	If field stats with this string then include the recipient
ends with	ends	If fields ends with this string then include the recipient
is greater than	greater	If field is a number that is greater than value then include the recipient
is less than	less	If field is a number that is less than value then include the recipient

```
"filter":{
  "match": "any",
  "rules": [
    {
      "field": "Fornavn",
      "operation": "is",
      "value": "chuck"
    },
    {
      "field": "Fornavn",
      "operation": "not",
      "value": "jack"
    },
    {
      "field": "Fornavn",
      "operation": "contains",
      "value": "cool"
    }
  ]
}
```

segmentation

Segmentation is a json array that tells the system how to split/limit who to send to.

Name	Description
count	If set how many recipients to send to
percent	If set how big a percentage of the segment should be used
exclude	List of previous batches

Sending to 10% of a list

```
"segmentation":{
  "percent": 10
}
```

Sending to the rest of the list

```
"segmentation":{
  "percent": 100,
  "exclude": [ 12345 ]
}
```

Responses

A couple of examples of responses

success

```
{
  "stat": {
    "sendtime": "17-12-2014 08:42:24",
    "buffered": 2,
    "received": 0,
    "rejected": 0
  },
  "details": {
    "sendtime": "17-12-2014 08:42:24",
    "batchid": "310701",
    "state": "DONE"
  },
  "status": 200
}
```

wrong list id

```
{
  "status":400,
  "message":"List ID 12 not found"
}
```

GET /messages/{page}.{format}

Get list of the newest batches including future waiting batches

```
curl 'http://api.zumosms.com/v2/messages/{page}.json?apikey=[apikey]' -X GET
```

```
curl 'http://api.zumosms.com/v2/messages/{page}.xml?apikey=[apikey]' -X GET
{
  "messages": [
    {
      "batchid": "2229409",
      "time": "23-03-2015 10:59:19",
      "state": "DONE",
      "sender": "TEST",
      "msg": "Hello world",
      "protocol": "SMTP"
    },
    {
      "batchid": "2229405",
      "time": "23-03-2015 10:50:16",
      "state": "DONE",
      "sender": "TEST",
      "msg": "Hello world",
      "protocol": "REST"
    }
  ],
  "status": 200
}
```

GET /message/{batchid}/{page}.{format}

Get message details based on batch id from POST

If page is 0 only summary is returned

```
curl 'http://api.zumosms.com/v2/message/{batchid}/{page}.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/message/{batchid}/{page}.xml?apikey=[apikey]' -X GET
```

page = 0

```
{
  "stat": {
    "sendtime": "17-12-2014 08:42:24",
    "buffered": 0,
    "received": 1,
    "rejected": 1
  },
  "details": {
    "sendtime": "17-12-2014 08:42:24",
    "batchid": "310701",
    "state": "DONE"
  },
  "status": 200
}
```

page = 1

```
{
  "message": [
    {
      "msgid": "d2dc8158a56d5bfb201c357e950bf9e6",
      "status": "rejected",
      "recipient": "412345678",
      "statusreason": "Unknown subscriber"
    },
    {
      "msgid": "aeb4205213393c0da008b44f527037e7",
      "status": "received",
      "recipient": "4587654321",
      "statusreason": ""
    }
  ],
  "stat": {
    "sendtime": "17-12-2014 08:46:33",
    "buffered": 0,
    "received": 1,
    "rejected": 1
  },
  "details": {
    "sendtime": "17-12-2014 08:46:33",
    "batchid": "310713",
    "state": "DONE"
  },
  "paging": {
    "current": 1,
    "pages": 1
  },
  "status": 200
}
```

PUT /message/refund/{lookup-id}.{format}

Refunds a message. If a message is specified in the payload, the user will be notified about the refund (The notification will be sent from the users account).

```
curl 'http://api.zumosms.com/v2/message/refund/{lookup-id}.json?apikey=[apikey]' -X
PUT -H "Content-Type: application/json" -d @file.json
curl 'http://api.zumosms.com/v2/message/refund/{lookup-id}.xml?apikey=[apikey]' -X
PUT -H "Content-Type: text/xml" -d @file.xml
```

Payload file.json

```
{
  "refund":{
```

```

    "message": "Your message will be refunded"
  }
}

```

On success, the response is

```

{
  "status": 200,
  "statusreason": "Message is now in queue for refund."
}

```

9. Query interface

GET /inbox/{page}.{format}

Query incoming messages

```

curl 'http://api.zumosms.com/v2/inbox/{page}.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/inbox/{page}.xml?apikey=[apikey]' -X GET

```

Fields

You can use the following GET parameter when searching for messages

Name	Type	Description
recipient	string	Search for messages with specific recipient
sender	String	Search for messages with a specific sender
starttime	Date dd-mm-YYYY HH:MM	Search time interval start
endtime	Date dd-mm-YYYY HH:MM	Search time interval end
msgid	String	Search for messages with msgid
message	String	Wildcard search in text
keyword	String	Search for message to specific keyword.

response

```

{
  "query": [
    {
      "lookupid": "1424185035-66417",
      "msgid": "4ab1a952e2645f4d39e06e26c9c220d6",
      "recipient": "+45609940305097",
      "sender": "4512345678",

```

```

        "logtime": "17-02-2015 15:57:15",
        "message": "Hmn"
    }
],
"paging": {
    "pages": 1,
    "offset": 10,
    "current": 1
},
"stats": {
    "total": 1
},
"status": 200
}

```

GET /inbox/lookup/{lookupid}.{format}

Get details for a single incoming message

```

curl 'http://api.zumosms.com/v2/inbox/lookup/{lookupid}.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/inbox/lookup/{lookupid}.xml?apikey=[apikey]' -X GET

```

response

```

{
    "query": {
        "msgid": "4ab1a952e2645f4d39e06e26c9c220d6",
        "recipient": "+45609940305097",
        "sender": "4512345678",
        "logtime": "17-02-2015 15:57:15",
        "message": "Hmn",
        "keyword": "HMN",
        "function": "liste"
    },
    "status": 200
}

```

GET /outbox/{page}.{format}

Query outgoing messages

```

curl 'http://api.zumosms.com/v2/outbox/{page}.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/outbox/{page}.xml?apikey=[apikey]' -X GET

```

Fields

You can use the following GET parameter when searching for messages

Name	Type	Description
recipient	string	Search for messages with specific recipient
sender	String	Search for messages with a specific sender
starttime	Date dd-mm-YYYY HH:MM	Search time interval start
endtime	Date dd-mm-YYYY HH:MM	Search time interval end
price	Integer	Search for messages that has price
price_expression	Enum (<, >, =)	How to search for price
batchid	Integer	Search for messages with batchid
msgid	String	Search for messages with msgid
message	String	Wildcard search in text
status	String	Search for messages with specific status
mcc	String	Either a single [http://en.wikipedia.org/wiki/Mobile_country_code] or a comma seperated list

response

```
{
  "query": [
    {
      "lookupid": "1424185035-66418",
      "msgid": "852e453d32e8459af090a3019fa9f9f6",
      "batchid": "659177",
      "status": "received",
      "statusreason": null,
      "recipient": "4512345678",
      "sender": "TEST",
      "logtime": "2015-02-13 10:32:44",
      "protocol": "REST",
      "message": "Hello world!",
      "statuscode": "0",
      "mcc": "238",
      "splitcount": "1"
    }
  ],
}
```

```
{
  "lookupid": "1424185035-66419",
  "msgid": "cb95abc5027ff19a907662c614f4f475",
  "batchid": "673449",
  "status": "received",
  "statusreason": null,
  "recipient": "4512345678",
  "sender": "TEST",
  "logtime": "2015-02-13 14:35:29",
  "protocol": "REST",
  "message": "æøå",
  "statuscode": "0",
  "mcc": "238",
  "splitcount": "1"
},
{
  "lookupid": "1424185035-66420",
  "msgid": "4ca01538b55c2ee68e531eb04773966a",
  "batchid": "679733",
  "status": "received",
  "statusreason": null,
  "recipient": "4512345678",
  "sender": "TEST",
  "logtime": "2015-02-13 15:43:45",
  "protocol": "REST",
  "message": "æøå",
  "statuscode": "0",
  "mcc": "238",
  "splitcount": "1"
}
],
"paging": {
  "pages": 1,
  "offset": 10,
},
"stats": {
  "total": 3,
  "splitcount": 3
},
"status": 200
}
```

GET /outbox/lookup/{lookupid}.{format}

Get details for a single message

```
curl 'http://api.zumosms.com/v2/outbox/lookup/{lookupid}.json?apikey=[apikey]' -X GET
```

```
curl 'http://api.zumosms.com/v2/outbox/lookup/{lookupid}.xml?apikey=[apikey]' -X GET
```

response

```
{
  "query": {
    "msgid": "7076fc592b59097b3c48c1f6e3321b6d",
    "batchid": "308605",
    "status": "received",
    "statusreason": "",
    "recipient": "4512345678",
    "sender": "Test",
    "logtime": "2014-12-15 09:03:47",
    "protocol": "REST",
    "message": "Hello world",
    "returndata": "",
    "statusurl": "",
    "statusresponse": "",
    "statuscode": "0",
    "push_price": "0"
  },
  "status": 200
}
```

10. Stats

GET /stats/{type}/{aggregate}.{format}

The stats system is designed to give you fast access to aggregated data.

⚠ Stats are calculated and are they are not realtime

```
curl 'http://api.zumosms.com/v2/stats/{type}/{aggregate}.json?apikey=[apikey]' -X GET
```

```
curl 'http://api.zumosms.com/v2/stats/{type}/{aggregate}.xml?apikey=[apikey]' -X GET
```

Type

Choose what type of messages you want to search in

Name	Description
inbox	Get stats for incoming messages
outbox	Get stats for outgoing messages

Aggregate

Choose what aggregation level you want to get stats for

Name	Description
hourly	Get hourly stats
daily	Get daily stats
weekly	Get weekly stats
monthly	Get monthly stats

Fields

You need to select what you want to search for in the stats system

Name	Type	Description
group	String	Search group id
starttime	Date dd-mm-YYYY HH:MM	Search time interval start
endtime	Date dd-mm-YYYY HH:MM	Search time interval end

The search groups are pre calculated stats

Type	Group	Description
inbox	sum	Total number of messages received
inbox	keyword:number	Number of received messages per keyword/number
outbox	sum	Total number of messages sent
outbox	mcc:status	Number of sent messages per country/status

If you have admin rights you can select the base aggregation level by using the following options

Name	Type	Description
advanced	String	Advanced key that gives you extended access to the stats interface
type_1	enum (user, mcc)	Select base type
value_1	String	Value to search for

examples

outbox/daily

```
curl 'http://api.zumosms.com/v2/stats/outbox/daily.json?apikey=[apikey]&starttime=15-02-2015 00:00:00&endtime=18-02-2015 23:59:59&group=sum' -X GET
{
```

```

"stats": [
  {
    "year": "2015",
    "month": "2",
    "day": "23",
    "timestamp": "2015-02-23 00:00:00",
    "type_1": "user",
    "value_1": "7405",
    "type_2": "sum",
    "value_2": "sum",
    "counter": "1"
  },
  {
    "year": "2015",
    "month": "2",
    "day": "25",
    "timestamp": "2015-02-25 00:00:00",
    "type_1": "user",
    "value_1": "7405",
    "type_2": "sum",
    "value_2": "sum",
    "counter": "4"
  }
],
"status": 200
}

```

outbox/monthly

```

curl 'http://api.zumosms.com/v2/stats/outbox/monthly.json?apikey=[apikey]
&starttime=01-01-2015 00:00:00&endtime=28-02-2015 23:59:59&group=sum' -X GET

```

```

{
  "stats": [
    {
      "year": "2015",
      "month": "2",
      "timestamp": "2015-02-01 00:00:00",
      "type_1": "user",
      "value_1": "7405",
      "type_2": "mcc:status",
      "value_2": "0:queued",
      "counter": "1"
    },
    {
      "year": "2015",
      "month": "2",

```

```

        "timestamp": "2015-02-01 00:00:00",
        "type_1": "user",
        "value_1": "7405",
        "type_2": "mcc:status",
        "value_2": "0:rejected",
        "counter": "1"
    },
    {
        "year": "2015",
        "month": "2",
        "timestamp": "2015-02-01 00:00:00",
        "type_1": "user",
        "value_1": "7405",
        "type_2": "mcc:status",
        "value_2": "238:received",
        "counter": "4"
    }
],
"status": 200
}

```

inbox/monthly

```

curl 'http://api.zumosms.com/v2/stats/inbox/monthly.json?apikey=[apikey]
&starttime=01-01-2015 00:00:00&endtime=28-02-2015 23:59:59&group=keyword:number' -X GET
{
    "stats": [
        {
            "year": "2015",
            "month": "2",
            "timestamp": "2015-02-01 00:00:00",
            "type_1": "user",
            "value_1": "7405",
            "type_2": "keyword:number",
            "value_2": "HENRIK:1272",
            "counter": "1"
        }
    ],
    "status": 200
}

```

POST /stats/inbox.{format}

Ingest data into the stats system. Requires an internal advanced key.

The max batch size to post is 10000

```

curl 'http://api.zumosms.com/v2/stats/inbox.json?apikey=[apikey]' -X GET

```

```
curl 'http://api.zumosms.com/v2/stats/inbox.xml?apikey=[apikey]' -X GET
{
  "stats": [
    {
      "code": "1272",
      "keywordid": "108406",
      "logtime": "2015-07-06 15:06:23",
      "materialized_path": "15m/49p",
      "msgid": "1436187983-48305",
      "topbruger": "1498",
      "userid": "7405"
    },
    {
      "code": "+4512345678",
      "keywordid": "1234",
      "logtime": "2015-07-06 15:06:31",
      "materialized_path": "abc",
      "msgid": "1436187991-48309",
      "topbruger": 0,
      "userid": "7405"
    },
    {
      "code": "1272",
      "keywordid": "107543",
      "logtime": "2015-07-06 15:06:32",
      "materialized_path": "1xg",
      "msgid": "1436187992-48313",
      "topbruger": "0",
      "userid": "7405"
    }
  ]
}
```

POST /stats/outbox.{format}

Ingest data into the stats system. Requires an internal advanced key.

The max batch size to post is 10000

```
curl 'http://api.zumosms.com/v2/stats/outbox.json?apikey=[apikey]' -X GET
curl 'http://api.zumosms.com/v2/stats/outbox.xml?apikey=[apikey]' -X GET
{
  "stats": [
    {
      "logtime": "2015-07-06 00:00:24",
      "materialized_path": "1s9/4j1",
      "mcc": "238",
      "msgid": "f151c342147599d3955c485f98a10d91",
```

```
    "payout": null,  
    "price": "",  
    "splitcount": "1",  
    "status": "received",  
    "topbruger": "2313",  
    "userid": "5869",  
    "group": "external"  
  },  
  {  
    "logtime": "2015-07-06 00:00:39",  
    "materialized_path": "6lp",  
    "mcc": "238",  
    "msgid": "fe8f9e28811bf0f21f95eefbc59c4142",  
    "payout": null,  
    "price": "",  
    "splitcount": "1",  
    "status": "received",  
    "topbruger": "0",  
    "userid": "8557",  
    "group": "internal"  
  }  
]  
}
```