24 Oct 2022

SMS SPECIFICATION 1.29 Advanced Copyright eCommunicate

Reg. No: CK2004/107057/23
VAT No.: 4510221395

Ecommunicate
102 Lyndhurst Road
Lyndhurst

Table of Contents

1. Overview

This document provides the technical specification by which all external content providers should abide while sending SMS.

Developers using the API URL to send SMSs namely either **https://api.ecommunicate.co.za**

To view the SMSs sent you can use the websites: **https://messaging.ecommunicate.co.za**

However the IP address may change so please use the URLs above.

Note: Please ensure you allow a fall back to HTTP in case there is any delay regarding the https certificate renewal.

We have followed in most cases the REST architecture as described in the Wikipedia link. **http://en.wikipedia.org/wiki/Representational_state_transfer**

"The architectural properties affected by the constraints of the REST architectural style are:

- Performance - component interactions can be the dominant factor in user-perceived performance and network efficiency.
- Scalability to support large numbers of components and interactions among components"

This document provides a detailed description of the URL addresses and keywords required in order to communicate with the message gateway.

Please see section 18, point c on how to prevent duplicate SMSs being sent.

2. Document History

REVISION NO	REVISION DATE	CHANGES
1.0 Beta	1 May 2015	
2.0 Beta	6 May 2015	Changes in Failure Recovery. Added Appendix 1 and 2
3.0 Beta	7 May 2015	Modified JSON (Added Settings parameter) Changes in Call Back URL Sections.
4.0 Beta	15 May 2015	Changes in Tracking Status and Query Reply Section
5.0 Beta	15 June 2015	Added Username Password limitation Section. Supports Message concatenation Added User and Reseller View
6.0 Beta	19 June 2015	Added Http Get to send SMS Invalid URL section. Explained Message concatenation Changed input time date format
7.0 Beta	2 Jul 2015	Added HTTP POST functionality for tracking SMS status, Query Reply, get

		Balance and get session ID, Added Dynamic Emails
8.0 Beta	17 Jul 2015	Added Blacklist functionality
9.0 Beta	21 Jul 2015	Modified Tracking Status Limit section, Changed time parameter in response to completion time, Changes in Query reply section
10.0 Beta	30 Jul 2015	Added includelastSentSms parameter in Query Reply section
11.0 Beta	5 Aug 2015	Changed includelastSentSms to includeSentSms
12.0Beta	7 Aug 2015	Added other considerations section Changed username parameter to userId
13.0 Beta	12 Aug 2015	Allowed lowercase and camel case parameters.
14.0 Beta	14 Aug 2015	Added scheduled SMS functionality, tracking scheduled SMS and deleting them. Section 18 ,19 and 20
15.0 Beta	28 Aug	Combined tracking scheduled SMS to Track SMS part
16.0 Beta	03 Sep	Added SMS Flow Section, Changed Validation errors and status messages.
17.0 Beta	18 Sep	Removed snapshots of UI, changed invalidMessages parameter, Removed scheduled SMS delay, removed repeat all replies option
18.0 Beta	23 Sep	Added show ctid parameter in trackingstatus , changed response object for invalid messages
19.0Beta	25 Sep	Added showfinalstatus parameter in tracking SMS status section

20.0Beta	1 Oct	Added UI section
21.0Beta	15 Oct 2015	Added info in special characters section
22.0 Beta	20 Oct 2015	Added getServerTime Section, Changed failure Recovery, Changed latest sendtime section
23.0 Beta	27 Oct 2015	Added information of serverReceiptTime and lastStatusChange parameter
24.0 Beta	30 Oct 2015	Added followMarketingRules Parameter Section
25.0 Beta	7 Dec 2015	Fixed JSON request examples
26.0 Beta	17 Dec 2015	Added information of adding new line in Text Message
27.0 Beta	6 Jan 2016	Added information about gctid and ctid
28.0 Beta	11 Jan 2016	Added information about batches
29.0 Beta	20 Jan 2016	Changes in ctid Section.
30.0 Beta	21 Jan 2016	Change in maximum batch size
31.0 Beta	28 Jan 2016	Changes in: SMS processing Sections Changed batches and message parameter to messages and message Text. Changed Error and Status Codes
32.0 Beta	29 Jan 2016	Changed trackSMS to trackingMessageStatus
33.0 Beta	1 Feb 2016	Changes in page no. 23 , 24,19
34.0 Beta	1 Feb 2016	Changes on Page no. 16,23 ,24,25,27 Changed max batch size to 10000
35.0 Beta	3 Feb 2016	Page no :21 (Change in schedule SMS date format)

36.0 Beta	8 Feb 2016	Change in pg 16(Changed max batch size to5000) Removed trackingProcessedMessage URl Added function to trackStatusmessage using gctid and gstid -Pg 24,25 Changed date format to delete scheduled message-Pg 35
37.0 Beta	15 Feb 2016	Changes in: Special Characters Section on pg 41
38.0 Beta	26 Feb 2016	Changed max batch size to 2000
39.0 Beta	1 Mar 2016	Changed max batch size to 1000 - see pg16 and increased concurrent connections to 10 - see pg 18
40.0 Beta	4 Mar 2016	Change in Pg -50 Get balance Section
41.0 Beta	18 Mar 2016	Change in pg-26 (added identifier param in receipt response) Pg- 48(added identifier , uuid , ctid) param in reply response
42.0 Beta	1 Apr 2016	Change in pg 41(change uuid to reply id) Change in pg 43,44 (added latestSendTime ,nextDayEarliestSendTime parameter)
43.0 Beta	11Apr 2016	Change in pg 37 (added Delete Scheduled SMS By ctids), Change in pg 43 (Sending Time Parameters)

44.0 Beta	19 Apr 2016	Change in pg 43(Sending Time Parameters) Change in Appendix1 - Pg 63
45.0 Beta	25 Apr 2016	Added SSL URL support.
1.1	21 June 2016	Sending Unicode Messages (pg-19)
1.2	9 Aug 2016	Added Unicode message Section (pg-46)
1.3	13 Dec 2016	Changed in sending new line in SMS Added information on sending tab space in SMS
1.4	24 Feb 2017	Added sentTime parameter in tracking sms response
1.5	3 Aug 2017	Unique ctid check, Testing Json
1.6	15 Sept 2017	Parameter Isunicode changed to unicode
1.7	6 Dec 2017	Added note to use /n in text by not breaking line and Message Rejected For Duplication
1.8	7Dec 2017	Added Get Unsubcribed Destination Api
1.9	6Feb 2018	Added using bulk SMS format to send a single SMS
1.10	5 May 2018	Added new Attribute to have businessUnit per message
1.11	7 May 2018	Added new Attribute to have businessUnit per batch
1.12	5 July 2018	Changed the URL as per new domain name
1.13	13 Aug 2018	Added Reply Call Back URL format
1.14	22 Aug 2018	Added Track Reply Report

1.15	14 Sept 2018	Added new attribute checkCreditsLater per batch
1.16	29 Oct 2018	Changed the parameters to be sent to ReceiptCallBackUrl
1.17	15 Nov 2018	Changed the details of validation error code- 026 and defaults set for checkCreditLater parameter while sending sms
1.18	19 Nov 2018	Added information on how to send URL in SMS
1.19	28 Nov 2018	Added additional information on Business Unit on how to add businessUnit at 3 levels: Batch, Message, Recipient
1.20	26 July 2019	Added recipients column while tracking message status
1.21	23 Aug 2019	Added Sending SMS's with EarliestSendTime and LatestSendTime
1.21	23 Aug 2019	Added Sending SMS's with EarliestSendTime and LatestSendTime
1.22	23 Aug 2019	Added Sending SMS's with EarliestSendTime and LatestSendTime
1.23	25 Aug 2019	Added DSM
1.24	30 Dec 2020	Added priority

1.25	22 Mar 2021	Added showCtId in Get and Post Request for receipt tracking url
1.26	12 May 2021	Added API to track sms sent to a recipient within a date range.
1.27	28th March 2021	Added OTP flag
1.28	9 Oct 2022	Added two conditions for clients requiring the OTP SMS service. A) If the OTP Token is invalid the status code is 480 and the SMSs will then be sent via the default queue on the primary SMS gateway and not by the high priority OTP queue. B) For those clients that require 24/7 uptime and who use the backup gateway: If the status code is set to 502 please send the SMSs via the backup gateway.
1.29	24 Oct 2022	Added otpToken to HTTP/S header

3. Getting Started

This technical document explains eCommunicate HTTP API to send and receive single/bulk messages.

Before using the SMS API the client must have a registered account. To register with our API log on to HTTP/S **https://messaging.ecommunicate.co.za** or **http://messaging.ecommunicate.co.za** For registration you will have to use an authorized email address called a emailid in this spec, which will be used for login and for sending SMS along with password and clientref parameter. You can log in to your account anytime to check delivery receipts and replies sent to you; also you can check credits left on your account.

4. Invalid URL

In general a URL may contain any of the following characters:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-._~:/?#[]@!$&'()*+,;=

But some of these characters come under reserved characters.
RESERVED CHARACTERS: **$ & + , / : ; = ? @ # % < >**

A URL becomes invalid if it contains one of the reserved characters**:**

If parameters in URL have reserved characters included then they must be used in percentage encoded format.

Symbols	Percentage Encoded Symbol
#	%23
$	%24
?	%3F
/	%2F
:	%3A
\	%5C
%	%25
&	%26
+	%2B
;	%3B
=	%3D

@	%40
<	%3C
>	%3E
.	%2E

- For example:

https://api.ecommunicate.co.za/sms/getBalance?emailid=testaccount@test.com&password=de#f&clientref=ghi

In above URL the word password has # and word emailid have a @ and a . character that must be percentage encoded as follows:

https://api.ecommunicate.co.za/sms/getBalance?emailid=testaccount2%40test%2Ecom&password=de%23f&clientref=ghi

Click on the following URL to encode your message properly:
http://meyerweb.com/eric/tools/dencoder/

To understand more about how URL encoding works follow the links below

http://www.tutorialspoint.com/html/html_url_encoding.htm

https://en.wikipedia.org/wiki/GSM_03.38

5. Other Considerations

- All parameter‘s used in this Spec are in either lowercase or lower Camel Case .

Camel Case :
Camel Case is a naming convention in which a name is formed of multiple words that are joined together as a single word with the first letter of each of the multiple words capitalized so that each word that makes up the name can easily be read in **lower camel case** the first letter of the first name is lower case.

- If parameter name's are not used correctly then user will receive an HTTP BAD request (400) error in case of HTTP POST requests and in HTTP GET requests user may INVALID PARAMETERS error.

- Please use **http://codebeautify.org/jsonvalidate** or **https://jsonformatter.curiousconcept.com/** to check if JSON syntax is correct while doing HTTP POST request.

6. SMS Processing

Our application supports sending of one or multiple messages.
Each message undergoes two phases:

a. Validation Phase:

i. This phase checks for authentication of the client.

ii. This phase will assume that clients have sufficient credits and all cell numbers submitted are correct.

iii. Check if gctid parameter provided is duplicate(This check is performed on daily basis i.e. the gctid parameter should be unique for today, it can be once repeated on next day).

When a message passes the validation phase we send a response back to the client that message(s)has been accepted for processing otherwise message(s) will be rejected with a response containing the error message.

All error messages that will be sent in response during validation phase are contained under Appendix 1: Validation Error Messages

b. Delivery and sending Phase:

i. Checks if client have sufficient credits.

ii. Checks if ctid parameter provided is duplicate (This check is performed on daily basis i.e. the ctid parameter should be unique for today, it can be once repeated on next day).

iii. This will check if all the recipients are valid and if it’s not blacklisted and give back credits if they are blacklisted or invalid .

iv. Checks for duplicate recipients and reject those messages and gives back credits.

v. Sends message(s).

vi. On tracking message status it sends back status message for each sent SMS.

vii. If no delivery receipt has come back after the latest time the MNO checks for as delivery receipt then the status will be set to STALE

viii. All unsent messages can be viewed either on website or by the using trackingMessageStatus URL.

We want the system to be as asynchronous as possible for performance enhancements and thus we have reduced processing in the validation phase. For more detail on asynchronous processing visit **https://en.wikipedia.org/wiki/Asynchronous_communication**

7. User Interface

User can have one of the two roles:

- Parent-Reseller: Reseller can add children under them and assign credits to the child's. Parent reseller has the authority to monitor all SMS sent and received by child-Resellers under him.

- Child-Reseller: Child User can send SMS and monitor sent and received SMS generate his reports via website.

a. Login Page

User can login by using his credentials on **http://messaging.ecommunicate.co.za**.
If a user tries to login using wrong credentials for three times then he will be blocked.

Please contact ecommunicate support to unblock your account in that case.

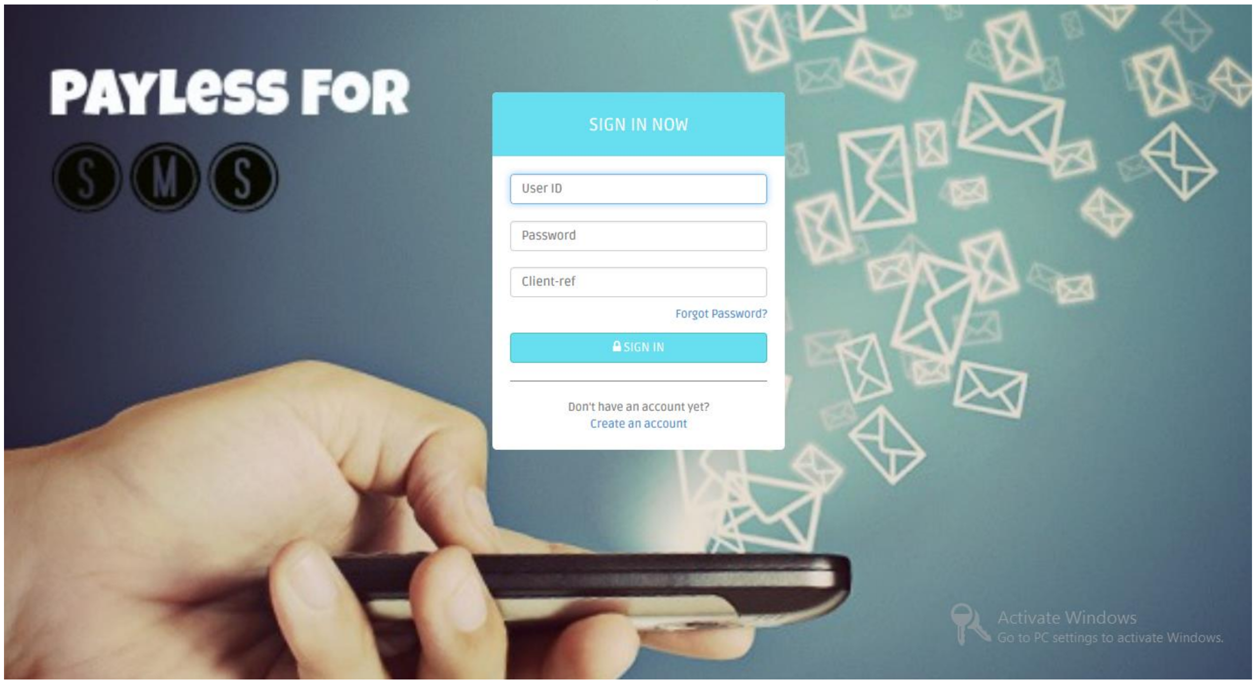


Figure 1 - Login

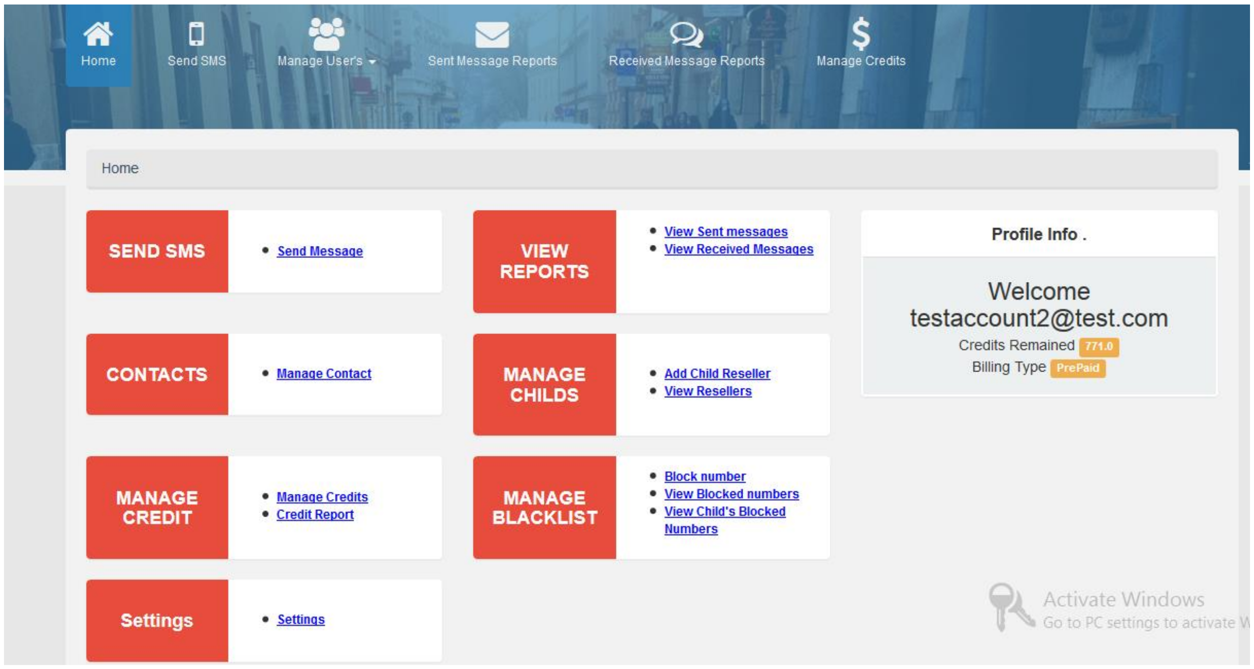


Figure 2 – Home Page

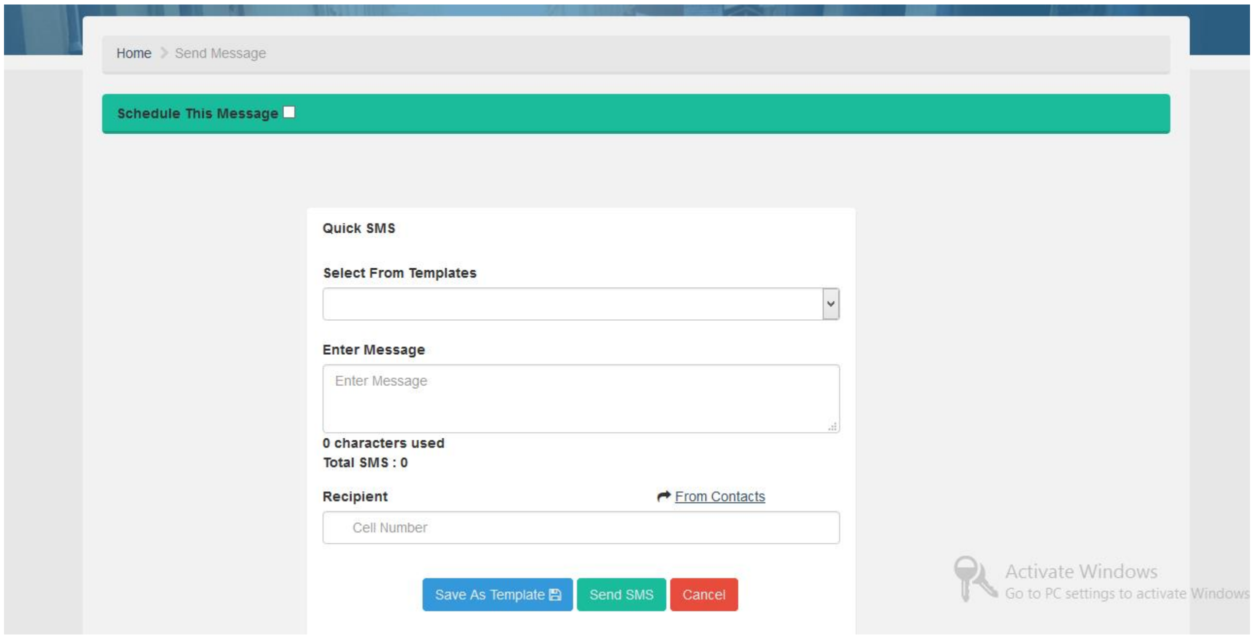


Figure 3 – Send SMS

8. Sending Message

Following is our SMS sending URL which is a JSON REST WEBSERVICE request:

HTTP/S : **api.ecommunicate.co.za**

Our API can be used to send three types of message:

- Sending a single message
- Sending single message to multiple recipients
- Sending multiple messages to multiple recipients.

8.1 OTPs, Tokens and Backup SMS gateway

For those clients sending OTPs (One Time Pins) who require the pins to be delivered within 5 to 10 seconds or less, we have created a token that needs to be sent with the SMS. Please email support@ecommunicate.co.za to receive your token. This token should be sent in the HTTP header with your SMS.

We check that the token sent with the SMS is that same as was provided to your company (each company's token is unique) and we also check if the token is invalid. Should we find that a company is misusing the OTP queue, for example sending marketing SMSs, then we will invalidate the token and the SMSs will go on the standard queue. Should the token be valid automatically the SMS is set on the OTP queue and so the queue name does not need to be mentioned in the SMS request. The SMS is also automatically set to priority, so the priority keyword does not need to be used.

A) If the OTP Token is invalid the **status code returned is 480 and the SMSs will then be sent via the default standard queue on the primary SMS gateway and not by the high priority OTP queue**.

B) For those clients that require 24/7 uptime and who use the backup SMS gateway: **If the status code is set to 502 please send the SMSs via the backup gateway**. This can happen should the primary SMS gateway be sending SMSs slower than usual and in that case the status code will be changed from the default to 502. Should the status code be changed to 502 please ensure that your system is configured to send the SMSs via the backup gateway, until the status code is set back to the default value of 100.

```
Headers:
Content-Type : application/json
otpToken : 7C84************

HTTPS POST:   https://api.ecommunicate.co.za/sms/json
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"},
"ctid":"250420160639",
"to":"27800000000",
"messageText":"hello"
}
```

8.2 Sending Message Using HTTP GET:

To quick send one text message to one or more destinations without using JSON you can use our HTTP GET URL.

Http Get approach must be used for testing purpose only. For sending bulk messages it is advised to use Sending multiple text messages to multiple destinations: option.

1) Sending one message:

For sending text message to one destination just provide the below URL with required parameters.

For example:
If your emailid is : testaccount@test.com then you will have to url encode " @" and "." as follows.

https://api.ecommunicate.co.za/sms/json?emailid=testaccount%40test%2Ecom&password= test123&clientref=test&to=0278300000&messageText=happy%20new%20year

Used Parameters:

i. emailid : email Id used while registration.
ii. password : password
iii. clientref : clientref
iv. to: mobile number of receiver. See Destination Address (to)
v. messageText : Text to be sent
vi. dsm : Set dsm as true to not save message on our server.
vii. priority: Set priority to true or false, when priority is set true the sms is transactional, and will be sent via premium route. When priority is set false the SMS is promotional, and will be sent via regular route, for example advertisement messages. Default is false.
Note: When priority is set true the SMS may cost more.
viii. OTP: Set otpToken in the header then the SMS is sent via a dedicated OTP queue.

2) Sending same text message to multiple recipients.

You can use our HTTP GET URL to send same text to multiple recipients by separating destination numbers with a comma (,) like the example below

https://api.ecommunicate.co.za/sms/json?emailid=testaccount%40test%2Ecom&password=test123&clientref=test&to=0278300002,0278300001&messageText=happy%20new%20year&

The HTTP GET method for sending message must be used while sending to few recipients as GET method is less secured then POST method.

POST method also provides additional functionality of sending multiple text messages to multiple destinations via JSON.

a. Sending Message Using HTTP POST

1) Sending a single text messages (SMS) to a single recipient using the bulk format.

Note: Priority flag should be string with value true or false.

Note: The terms messages and recipients can be used to send one message and to one recipient and to send multiple messages and to multiple recipients. This format should be used as the default format to send SMSs.

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2"},
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27737715199", "identifier":"John"}
],
"messageText":"hello123 "
}
]
}
```

2) Sending single text messages (SMS) to single recipients using the bulk format.

Note: Priority flag should be string with value true or false.
Note: otp flag should be string with value true or false.

The Client can send individual messages to individual recipients. With a single JSON request we can send one bulk SMS ; following is the basic JSON format for that:

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2"},
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27737715199", "identifier":"John"}
],
"messageText":"hello123 "
},
{
"ctid":"210420160917",
"recipients": [
{"to":"27832300002", "identifier":"Lisa"}
],
"messageText":"testing message "
}

]

}
```

3) Sending the same text message to multiple recipients.

Note: Priority flag should be string with value true or false.

The client can send one text message to multiple receivers as well; following is the basic JSON format for that:

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"},
"ctid":"210420160911",
"recipients": [
{"to":"27800000000", "identifier":"abc"},
{"to":"27800000001", "identifier":"123"}],
"messageText":"hello"
}
```

In the above format two extra parameters are added which are described below.

Used Parameters:

i. ctid: Client Tracking ID - this will be submitted by users to uniquely identify each message and it can be used to track status of messages sent in that message using destination number.
ii. recipients: This field is an array of recipients object that contains parameters 'to' (destination mobile number) and 'identifier'.

iii. to**:** recipient mobile number. see Destination
iv. identifier**:** This parameter could be the name of recipient or the reason the SMS was sent to the recipient e.g. New Audi user. Identifier is an optional parameter.

4) Sending multiple text messages to multiple destinations:

The client can send multiple text messages to multiple destinations. Basic JSON required for that is provided below:

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"},
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27737715199", "identifier":"John"},
{"to":"27737715190", "identifier":"Anna"}],
"messageText":"hello123 "
},
{
"ctid":"210420160917",
"recipients": [
{"to":"27737715197", "identifier":"Lisa"},
{"to":"27737715196", "identifier":"Ron"}],
"messageText":"Friendly reminder that you can receive your account balance by SMSing
the keyword BAL followed by your 7 digit account number to 12547 and 7 digit account
number to 12349"
}
]
}
```

This format has added messages parameter in it which is described below other used parameters are same as described above. The format has to be followed properly to send multiple text messages to multiple destinations.
Used Parameters:

i) messages: This parameter contains array of message.

message :
Each message comprises of at least one recipient and one message text along with ctid. A message can have multiple recipients in it.

The below example will be called as a message in this spec.

```
{
"ctid":"adsfawseva",
"recipients": [
{"to":"27737715197", "identifier":"abc"},
{"to":"27737715196", "identifier":"abc"}],
"messageText":" Friendly reminder that you can receive your account balance by SMSing
the keyword BAL followed by your 7 digit account number to 12547 and 7 digit account
number to 12349"
}
```

ii) gctid: client global tracking id- this will be submitted by users to uniquely identify all messages submitted in request ..

iii) ctid: client tracking id - this will be submitted by users to uniquely identify each message and it can be used to track status of messages sent in that message using destination number.

Note: While sending bulk messages the client must keep the JSON messages size to 1000 for good performance and speed. Each request must be sent after getting response of previous request.

5) Sending Unicode messages:

To send unicode messages use isUnicode parameter as follows:

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount",
"clientRef":"testaccount"
},
otp: false,
"unicode":"true",
"ctid":"201606210422",
"to":"27737725199",
"messageText":"Estimado/a cliente, Saiba que o seguro anual ou semestral na GA
oferece serviço de reboque gratuito ao veículo segurado (oferta limitada a zona de
Luanda). Ter seguro é bom. Ter GA é óptimo."
}
```

6) Adding New Line in a SMS

To add line break in text message user should use \n in case of JSON post and %0A in case of JSON get request.
For example

HTTP GET Request will be :

https://api.ecommunicate.co.za/sms/json?emailid=testaccount%40test%2Ecom&password=test123&clientref=test&to=0278300000&messageText=happy%20new%20year

This will be sent as :

happy
new year

HTTP POST Request will be :

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"client name",
"password":"password",
"clientref":"clientref"
},
"to":"27800000000",
"messageText":" Friendly reminder that you can receive your account balance by SMSing the keyword BAL followed by your 7 digit account number to 12547 \n and 7 digit account number to 12349"

}
```

This will be sent as :

```
Friendly reminder that you can receive your account balance by SMSing
the keyword BAL followed by your 7 digit account number to 12547
and 7 digit account number to 12349
```

Example to add text in new line

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"to":"27800000000",
"messageText":"hello \n test"

}
```

hello
test
hello\n test results John being on a new line.

Example to add blank new line

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"to":"27800000000",
"messageText":"hello \n \n John"

}
```

hello

John

Results contain John being on a new line and one separator line placed between hello and John.

Note*: Please Note we don't accept \r , if you send \r in the text it will be removed from text message.
If we want to send just \n and not break line then we need to use \\n to get \n in the text.

7) Adding Business Unit per message

To identify the business unit from where the message was sent from we have an attribute where the business unit can be set and thus those units can be reflected while fetching the reports from website

We have 2 ways of having Business Unit :

1.businessUnit per Recipient
2.businessUnit per Message

3.busineesUnit per Batch

businessUnit per Recipient:

Send request to : **https://api.payless4sms.co.za/sms/json**

Example to add business unit per recipient:

```
{
"settings" :{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"ctidunique":true
},
"messages":[{
"ctid":"fkjhkldfgjhkjdgghfhdfg",
"recipients": [
{"to":"28398892139", "businessUnit":"abc111"},
{"to":"28300892890", "businessUnit":"abc222"}],
"messageText":"Please check your account is in order. "
},
{
"ctid":"uweywiquyruietg",
"recipients": [
{"to":"28200892139", "businessUnit":"abc333"},
{"to":"28323456798", "businessUnit":"abc444"}],
"messageText":"Friendly reminder that you can receive your account balance by SMSing"
}
```

businessUnit per Message:

Send request to : **https://api.payless4sms.co.za/sms/json**

Example to add business unit per message:

```
{
"settings" :{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"ctidunique":true
```

```
},
"messages":[{
"ctid":"fkjhkldfgjhkjdgghfhdfg",
"recipients": [
{"to":"28398892139"},
{"to":"28300892890"}],
"messageText":"Please check your account is in order. ",
"businessUnit":"abc444"
},
{
"ctid":"uweywiquyruietg",
"recipients": [
{"to":"28200892139"},
{"to":"28323456798"}],
"messageText":"Friendly reminder that you can receive your account balance by
SMSing",
"businessUnit":"abc555"
}
```

busineesUnit per Batch

Send request to : **https://api.ecommunicate.co.za/sms/json**

Example to add business unit per batch:

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
,"businessUnit":"abcd",
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27111300002", "identifier":"John"}
],
"messageText":"Brioche is offering 3 pizzas for the price of 2 this Sunday till 2pm
Buy 2 pizzas and we'll give you a 3rd one free Get your 3 pizzas at Brioche this
Sunday. "
```

```
                 }

]

}
```

If businessUnit is provided at all the 3 levels: Batch, Message, Recipient then businessUnit at batch level would take precedence over Message and Recipient level. If businessUnit is not present at Batch level, but is present at Message level and Recipient level, then businessUnit at message level would take precedence over Recipient level businessUnit.

8) Adding Check Credits later per batch

To check the credits later while sending batch SMS instead of checking the credits while getting the response, we have an attribute where check credits later can be set to true and the batch containing checkCreditsLater attribute to true will be processed later.Adding check
By default, checkCreditsLater is set to false for prepaid users and true for postpaid users.

Send request to : **https://api.ecommunicate.co.za/sms/json**

Example to add checkCreditsLater per batch:

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
, "checkCreditsLater":true,
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27111300002", "identifier":"John"}
],
"messageText":"Brioche is offering 3 pizzas for the price of 2 this Sunday till 2pm
Buy 2 pizzas and we'll give you a 3rd one free Get your 3 pizzas at Brioche this
Sunday. "
}
]
}
```

If checkCreditsLater is set to true, you will get the following response:-

```
{
    "gstid": "06e9f458-6b6c-4b4e-8094-3498f2812852",
    "statusCode": "100",
    "statusDescription": "Message(s) Accepted For Processing",
    "completionTime": "Mon Sep 17 09:54:44 SAST 2018"
}
```

If checkCreditsLater is not specified or is set to false,then you will get following response:-
Note:- Suppose You have 1.0 credits.

```
{
"statusCode": "139",
"statusDescription": "Credits are less than total batch messages cost. Your Total
Batch Message Cost is 3 and Your Credits are 1.0.",
"completionTime": "Mon Sep 17 09:54:44 SAST 2018",
"eSTId": "67f1223e-7b4f-416a-994a-89099556750e"
}
```

9) Sending URL's in SMS

If user wants to send url with SMS. They need to send the request in UTF-8 format. If you are not getting / in url and its getting replaced with %2F, that means user is not sending request in UTF-8 format.

```
{
"settings":
{
"emailid":"testaccount2@gmail.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"messages":[{
"recipients": [
{"to":"27111300002", "identifier":"divya"}
],
"messageText":"hello123 URL http://www.bex.co.za/"
}
]
}
```

10) Sending SMS's with EarliestSendTime and LatestSendTime

To sending messages within particular time range in a day .For achieving this, need to add earliestSendTime and latestSendTime in parameter as shown below.

Description of these two parameters are as follow.

earliestSendTime : messages will not be sent before this time.

latestSendTime : messages will not be sent after this time.

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
,
"earliestSendTime":"8:00am",
"latestSendTime":"5:00pm",
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27111300002", "identifier":"John"}
],
"messageText":"Some text."
}
]
}
```

11) Don’t save message (DSM)

If dsm parameter is set to true then the messages which are marked with the dsm option will not get saved into the database. In this case we will save message body as “Do Not Save Message”.

Also we are sub masking the receiver number.

For e.g. If the receiver number is `27111300002,` we will submask that number like `xxxxxxx0002` and insert it into database.

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"dsm":"true"
}
,
"earliestSendTime":"8:00am",
"latestSendTime":"5:00pm",
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27111300002", "identifier":"John"}
],
"messageText":"Some text."
}
]
}
```

9. Session Id

To prevent having to put in the emailid, password and clientref or if the SMS has many cell numbers and it will be broken up into many connections to the SMS server, we recommend the client requests a sessionId. This should then be used throughout all communication with the SMS server

For getting a sessionID the user can use either GET or POST methods along with required parameters passed either in URL or as JSON. This will return a JSON response object containing a sessionId which will expire if session remains inactive for ten minutes. User has to enter this sessionId in the sessionId parameter.

a. Get SessionId using HTTP GET

To get sessionId by this method user provide authorized emailid, password and clientref in the format as given below.

https://api.ecommunicate.co.za/sms/getSessionId?emailid=testaccount2%40test%2Ecom&password=testaccount2&clientref=testaccount2

b. Get SessionId using HTTP POST

To get sessionId by this method user will have to provide a JSON with emailid, password and clientref in the format as given below :

Send request to : **https://api.ecommunicate.co.za/sms/getSessionId**

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

c. SessionId Response Object:

```
{
"sessionId":"9525be11-aaa3-4470-aaa0-d02739af1ca0",
"completionTime":"2015-05-21T21:22:26+05:30",
"statusCode":"001",
"statusDescription":"REQUEST_COMPLETED"

}
```

Response object will have following parameters:

i. sessionId: A unique identifier valid for sending SMS.
ii. completionTime: date/time.
iii. statusCode: status code. See - Appendix 1: Validation Error Messages
iv. statusDescription: Description of status code.

d. Using sessionId parameter

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"sessionId":"9525be11-aaa3-4470-aaa0-d02739af1ca0"
},
"to":"2780000000",
"messageText":"hello"
}
```

10. Client Tracking ID (ctid) and Client Global Tracking Id(gctid):

The ctid parameter (client tracking ID) is used to track all messages sent by the client. This must be unique for each message to get the full benefits of using the ctid.

As developers may create code using ctid's that are not unique and then leave the company and the systems are not maintained, we have set the ctid to not be unique by default.

If the client sends a message with a ctid that has been used before and `ctidunique` is set to true then the message will be accepted with the below response but it will not be sent and credits will be given back.

```
{
"gctid": "abcd",
"gstid": "3e99e8d9-50bf-469d-9fd4-d6c10cf5b152",
"statusCode": "100",
"statusDescription": " MESSAGE_ACCEPTED_FOR_PROCESSING ",
"completionTime":"2015-05-21T21:22:26+05:30"
}
```

You can track unsent messages due to duplicate ctid error using trackingMessageStatusURL or via website. We perform duplicate ctid check only for current day and thus while tracking messages using ctid you must provide correct date with it.

If system fails in between sending messages (system or power failure) and `ctidunique` is true then the client will be able to recover and resend their JSON requests with the ctids that they previously used and only unsent messages will be sent to the respective destination.

To ensure that messages should not be sent to any destination twice, the ctids along with destination number and message sending date will be used to track already sent SMS. See OverviewFailure Recovery

The gctid parameter will be used to track processed messages along with message submit date after they are submitted to our gateway. It should be uniquely generated to send messages; otherwise it will give Duplicate gctid error.

Note: Its developers responsibility to ensure that the ctid is unique otherwise they could get back "Accepted for Processing" and find out via the delivery receipts that the SMS's failed before sending as the ctid was not unique.

a. Using ctidunique (Boolean) parameter

This parameter is used to provide unique ctid check.

If it is set to true , **system will check for unique ctid in the batch as well as will check our database for messages sent with same ctid on that day**. In case they are not unique they will be rejected. If it is set to false or if it's not set , by default we will not check the uniqueness in ctids.

For example:
In the following json ctidunique is set to true , so ctid should be unique in the following ways:
1. It should not be same as other ctid in the json
2.It should not match with ctid used earlier on that day

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"ctidunique":true
},
"messages":[{
"ctid":"fkjhkldfgjhkjdgghfhdfg",
"recipients": [
{"to":"27111892139", "identifier":"John"},
{"to":"27111892140", "identifier":"Anna"}],
"messageText":"duplicate check "
},
{
"ctid":"uweywiquyruietg",
"recipients": [
{"to":"27111892149", "identifier":"Lisa"},
{"to":"27111892169", "identifier":"Ron"}],
"messageText":"Friendly reminder that you can receive your account balance by SMSing"
}

]
}
```

In case it fails in both condition the messages will not be sent and it will give Duplicate ctid.

```
{
    "gstid": "57ba9c57-7f37-44c1-8b1b-364755778f90",
    "statusCode": "001",
    "statusDescription": "Request Completed",
    "completionTime": "2017-08-03T18:04:24+0200",
    "receipts": [
        {
            "statusCode": "229",
            "statusDescription": "Duplicate ctid",
             "destination": "+27111892149",
            "ctid": "fkjhkldfgjhkjdgghfhdfg "
            "identifier": "test2",
ctidunique "sentTime": "2017-08-03T09:08:39.000+02:00"

                    }
    ]
}
```

Developers should set `ctidunique` to true which means they will have:

1) Extra reliability so that if SMSs fail they can resend and know with confidence that duplicate SMSs will not be resent

2) A method to prevent duplicates if they send each SMS with a unique ctid then if SMSs fails it means there was a duplicate

3) Delivery receipts are easily tracked.

The only way to prevent CTID being duplicated is to use a GUID. If the developer does not use a GUID then there is a risk that the CTID will not be unique and the SMSs will fail with the error "`Duplicate ctid` ". If a GUID is not used then we advise developers to not set the `ctidunique` value to true and to leave this parameter from their code as by default its set to false.

To understand a GUID see https://en.wikipedia.org/wiki/Universally_unique_identifier

11. System Tracking ID (stid) and Global stid (gstid):

The stid parameter is a system generated unique ID that will be sent in each response from the system either in sent SMS response or in error messages to keep track of message submitted from the client.

The gstid parameter is a system generated unique ID that will be sent in each response from the system either in sent SMS response or in error messages to keep track of all messages submitted in one request from the client.

12. Error system tracking Id (eSTId)

This will be unique system generated Id which will be returned in response object if the message fails system validation phase before sending message. If there are no errors in the validation phase and then the client will get response object with this parameter having null value.
This id can be used to keep track of errors that occurred after client submitted message. Clients can see failed messages on online account.

13. SMS Scheduling

SMS delivery can be scheduled on user provided date. To delay delivery of SMS, user should provide a date in format **YYYY-MM-DDTHH:mm:ssZ**.

The actual scheduled time for SMS can get a delay of 5 minutes. SMS can be scheduled by using scheduledSmsDate parameter. If user do not want to schedule SMS then this parameter should be ignored.

a. Using scheduledSmsDate parameter:

JSON format to include scheduledSMSDate while sending SMS:

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientRef":"testaccount2",
"scheduledSmsDate":"2016-04-22T09:36:26+02:00"
},
"ctid":"220420160105",
"to":"27800000000",
"messageText":"hello SMS is scheduled"
}
```

b. Scheduling SMS

If scheduledSmsDate parameter is used in settings block then all messages in input JSON will be scheduled

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientRef":"testaccount2",
"scheduledSmsDate":"2016-04-22T10:58:26+02:00"
},
"ctid":"042220160227",
"recipients": [
{"to":"27800000000", "identifier":"john"},
{"to":"27800000001", "identifier":"ria"}],

"messageText":" Friendly reminder that you can receive your account balance by SMSing
the keyword BAL followed by your 7 digit account number to 12547 and 7 digit account
number to 12349"

}
```

Another option can be to send few SMS and schedule few in one go.

c. Scheduling Few SMS

If scheduledSmsDate is used in the message block then only that message will be scheduled rest of all messages will be sent at the same time.

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{

"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"gctid":"042220160236"
},

"messages":[{
"ctid":"042220160237",
"recipients": [
{"to":"27768510326", "identifier":"john"},
{"to":"27737715190", "identifier":"ria"}],
"messageText":"hello",
  "scheduledSmsDate":"2016-04-25T11:10:26+02:00"
},
{
"ctid":"042220160238",
"recipients": [
{"to":"27737715196", "identifier":"ron"}
],
"messageText":"hello"
}

]
}
```

Response Object

If you have scheduled all messages then you will get the following response

```
{
"gctid": "abcd",
"gstid": "3e99e8d9-50bf-469d-9fd4-d6c10cf5b152",
"statusCode": "201",
"statusDescription": MESSAGE_BATCH_SCHEDULED ",
"completionTime":"2016-04-21T21:22:26+05:30"
}
```

You can only schedule message to a future date, otherwise you will get the following response

```
{
  "statusCode": "032",
  "statusDescription": "Scheduled Time must be greater than current time",
  "completionTime": "2016-04-25T13:41:17.000+02:00",
  "eSTId": "476f0674-b8ca-49ea-b971-0c92c784c026"
}
```

14. TRACKING MESSAGE STATUS

Status of sent messages can be tracked by using HTTP/S GET/POST or the UI.

To query status of any SMS from HTTP/S the client must hit the below provided URL and provide client authentication parameters (emailid ,password ,clientref) and other required parameters.

a. Track SMS using HTTP GET

- **Track SMS By time**
  User can query for SMS receipts sent in between a time period as follows:

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&startDate=2016-04-22 00:25:43&endDate=2016-04-23 18:25:43

startDate: Any receipts received after this date, including this date.

endDate: Any receipts received before this date, including this date.

Note :Format of input dates must be yyyy-mm-dd hh:mm:ss.

All dates will always be returned in YYYY-MMDDTHH:mm:ssZformat to enable multiple time zones.

- **TrackSMS by ctid / stid**

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount2%40test%2Ecom&password=testaccount2&clientref=testaccount&ctid=042220160238&sentDate=2016-04-22

or

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&stid=eff47ce1-9a88-46b2-8fd8-1f9bedc2b4e1&sentDate=2016-05-24

- **TrackSMS by gctid / gstid**

If user wants to get the entire set of delivery receipts for all the cell numbers that constitute the message then they can send gctid or gstid of that message along with sentDate and we will send receipts of all sent SMS of that message.

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&gctid=042220160236&sentDate=2016-04-22

or

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&gstid=700d7d5b-5437-4f09-8b4a-c76b06442633&sentDate=2016-04-22

- **Track SMS by stid and destination number**

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&stid=9100f436-0df2-4168-a490-b9f5d4ef096a&destination=27800000000

b. Track SMSs sent to recipient within date range

- **Track SMS by destination number and date range**
  https://api.ecommunicate.co.za/sms/getReceiptsForRecipient?userid=testaccount&password=testaccount2&clientref=testaccount2&recipient=27111456457&startDate=2021-03-01T00:46:19Z&endDate=2021-09-29T18:46:19Z

  Response :
  If no record exists : -

```
{
   "statusCode": "504",
   "statusDescription": "No Reference To Batch",
   "completionTime": "2021-05-12T14:25:12+0200"
}
```

  If record exists : -

```
{
   "gstid": "11a5c209-c5f6-458e-a7fc-bead80b8db7c",
   "statusCode": "001",
   "statusDescription": "Request Completed",
   "completionTime": "2021-05-12T19:17:47.000+05:30",
   "receipts": [
      {
         "id": 0,
         "stid": "15c32e91-aebf-4559-9b7b-34870009ae49",
         "statusCode": "600",
         "statusDescription": "Delivered to Recipient",
         "destination": "+27111234567",
         "ctid": "210420160916",
         "deliveredTime": "2021-03-01T09:05:38.000+05:30",
         "sentTime": "2021-03-01T09:05:35.000+05:30",
         "identifier": "John"
      }
   ]
}
```

c. Track SMS using HTTP POST

For tracking SMS using HTTP POST user must provide required parameters in JSON format as given below to **https://api.ecommunicate.co.za/sms/trackingMessageStatus**:

- **Track SMS By time**
  User can query for SMS receipts sent in between a time period as follows:

  Send request to : **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"startDate":"2016-04-22 00:48:16",
"endDate":"2016-04-23 00:48:16",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

startDate: Any receipts received after this date, including this date .
endDate: Any receipts received before this date, including this date.

Note :Format of input dates must be yyyy-mm-dd hh:mm:ss .

All dates will always be returned in YYYY-MM-DDTHH:mm:ssZ format to enable multiple time zones.

- **TrackSMS by ctid / stid**

Send request to : **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"ctid":"042220160238",
"sentDate":"2016-04-22",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

Or

```
{
"stid":"031b66aa-69f3-4310-af6a-c606a21ce3e9",
"sentDate":"2016-04-22",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

- **TrackSMS by gctid / gstid**

Send request to : **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"gctid":"042220160245",
"sentDate":"2016-04-22",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

Or

```
{
"gstid":"700d7d5b-5437-4f09-8b4a-c76b06442633",
"sentDate":"2016-04-22",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

- **Track SMS by stid and destination number**

Send request to : **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"stid":"031b66aa-69f3-4310-af6a-c606a21ce3e9",
"destination":"27737715196",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

Response Object

This will return a message status response object as follows:

```
{
    "gstid": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
    "gctid": "9c234op73b09b9",
    "statusCode": "224",
    "statusDescription": "REQUEST_COMPLETED",
    "completionTime": "2016-04-03T19:06:14.000+05:30",
    "receipts": [
        {
            "statusCode": "400",
            "statusDescription": "QUEUED_AT_MNO_FOR_ACCEPTANCE",
            "destination": "+277300000",
             "stid": "dd26dc95-0205-43fb-b723-f451c2ebf776"
           "identifier": "test2",
          "sentTime": "2017-02-20T09:08:39.000+02:00"
        }
    ],


}
```

Used Parameters:

i. gstid: Global system tracking Id .
ii. gctid: Global client tracking Id (returned if provided by user while sending SMS).
iii. stid : System tracking Id.
iv. statusCode: This represents the status code of the SMS.
v. eSTId: System generated unique Id to keep tracks of failures before message sending.
vi. StatusDescription: This provides description of the status code.

If client does not have enough credits to send that message then he will get following error message:

```
{
"gctid": "abcd",
"gstid": "3e99e8d9-50bf-469d-9fd4-d6c10cf5b152",
"statusCode": "137",
"statusDescription":"Insufficient Credit",
"completionTime":"2016-04-21T21:22:26+05:30"
}
```

While submitting messages if current time is 2015-02-13 11:59:59 pm, then messages may get saved with

2015-02-14 00:00:00 date and thus while tracking them you should use the next day date if the messages are not present with the previous day date.

c. Tracking Scheduled SMS

Status of scheduled messages can be tracked by using HTTP/S GET/POST or via UI.

Scheduled SMS can be tracked by using isScheduledSms parameter with any of the above provided URL's as follows.

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&isScheduledSms=true&stid=0a6a3de7-b974-42d0-af9e-9c23473b09b9&destination=278000...

Track scheduledSMS using JSON Post method as follows

Send request to : **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"stid":"031b66aa-69f3-4310-af6a-c606a21ce3e9",
"destination":"27737715196",
"isScheduledSms":"true"

}
```

Scheduled SMS Response Object

You will get the following response

```
{
    "gstid": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
    "gctid": "9c234op73b09b9",
     "statusCode": "224",
    "statusDescription": "REQUEST_COMPLETED",
    "completionTime": "2016-04-28T12:18:36+0200",
    "receipts": [
        {
            "statusCode": "201",
            "statusDescription": "Message Batch Scheduled",
            "stid": "98f73eec-5695-42e8-9023-897d1d558489",
            "messageText": "hello",
            "to": "2770000000",
            "scheduledDate": "2016-04-15 02:00:00.0"
        },
        {
           "statusCode": "201",
            "statusDescription": "Message Batch Scheduled",
            "stid": "98f73eec-5695-42e8-9023-897d1d558489",
            "messageText": "hello",
            "to": "2770000000",
            "scheduledDate": "2016-04-15 02:00:00.0"
        },
        {
           "statusCode": "201",
            "statusDescription": "Message Batch Scheduled",
            "stid": "98f73eec-5695-42e8-9023-897d1d558489",
            "messageText": "hello",
            "to": "2770000000",
            "scheduledDate": "2016-04-15 02:00:00.0"
        },
        {
            "statusCode": "201",
            "statusDescription": "Message Batch Scheduled",
            "stid": "98f73eec-5695-42e8-9023-897d1d558489",
            "messageText": "hello",
            "to": "2770000000",
            "scheduledDate": "2016-04-15 02:00:00.0"
        }

    ]
}
```

If there are no scheduled messages then you will get the following response

```
{
    "statusCode": "507",
    "statusDescription": "SCHEDULED_MESSAGES_NOT_FOUND",
    "completionTime": "2016-04-28T12:21:06+0200"
}
```

While submitting messages if current time is 2015-02-13 11:59:59 pm, then messages may get saved with

2015-02-14 00:00:00 date and thus while tracking them you should use the next day date if the messages are not present with the previous day date.

d. Using showctid Parameter

If user wants to include ctid of the message in the response then he can use showctid parameter as follows.

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&ctid=042220160237&sentDate=2016-04-22&showctid=true

Using showctid parameter in JSON Post

Send request to : **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"ctid":"042220160238",
"sentDate":"2016-04-22",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"showctid":"true"
}
```

Response with ctid :

```
{
    "gstid": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
    "gctid": "9c234op73b09b9",
    "statusCode": "001",
    "statusDescription": "REQUEST_COMPLETED",
    "completionTime": "2016-04-03T19:06:14.000+05:30",
    "receipts": [
        {
            "statusCode": "400",
            "statusDescription": "QUEUED_AT_MNO_FOR_ACCEPTANCE",
            "destination": "+277300000",
             "ctid": "042220160238"
            "identifier": "test2",
         "sentTime": "2017-02-20T09:08:39.000+02:00"
        }
    ],
}
```

e. Using showFinalStatus Parameter

To get all messages with final status only i.e. (Delivered , Undelivered ,Rejected) one can use showFinalStatus parameter. It can be used as follows

For example: Normal Response

```
{
    "gstid": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
    "gctid": "9c234op73b09b9",
    "statusCode": "224",
    "statusDescription": "REQUEST_COMPLETED",
    "completionTime": "2015-09-03T19:06:14.000+05:30",
    "receipts": [
        {
            "statusCode": "400",
            "statusDescription": "Queued at mno for acceptance",
            "destination": "+277300000",
             "ctid": "fdadsa-aadsfvsfdsad"
            "identifier": "test2",
            "sentTime": "2017-02-20T09:08:39.000+02:00"

        },
{
            "statusCode": "500",
            "statusDescription": "Message Sent",
            "destination": "+277300000",
             "ctid": "fdadsa-aadsfvsfdsad",
            "identifier": "test2"
            "latestStatusTime": "2015-09-03T19:12:27.000+02:00",
             "sentTime": "2017-02-20T09:08:39.000+02:00"

        },{
            "statusCode": "600",
            "statusDescription": "Delivered to recipient",
            "destination": "+277300000",
             "ctid": "fdadsa-aadsfvsfdsad",
           "identifier": "test2"
            "latestStatusTime": "2015-09-03T19:12:27.000+02:00",
          "sentTime": "2017-02-20T09:08:39.000+02:00"
        }


    ]


}
```

If showfinalstatus parameter is used then it will show only Delivered SMS from the above example as it's the only one which is in its final state.

Using showfinalstatus parameter by HTTP GET

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&gstid=700d7d5b-5437-4f09-8b4a-c76b06442633&sentDate=2016-04-22&showctid=true&showfinalstatus=true

Using showfinalstatus parameter by HTTP POST

Send request to : **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"ctid":"042220160238",
"sentDate":"2016-04-22",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"showctid":"true",
"showfinalstatus":"true"
}
```

Response

```
{
    "gstid": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
    "gctid": "9c234op73b09b9",
    "statusCode": "001",
    "statusDescription": "REQUEST_COMPLETED",
    "completionTime": "2015-09-03T19:06:14.000+05:30",
    "receipts": [
       {
            "statusCode": "600",
            "statusDescription": "Delivered to recipient",
            "destination": "+277300000",
             "ctid": "042220160238",
            "identifier": "test2"
            "latestStatusTime": "2015-09-03T19:12:27.000+02:00",
"sentTime": "2017-02-20T09:08:39.000+02:00"

}

    ]


}
```

f. ServerReceiptTime and LatestStatusTime Parameter

This parameter holds the time at which SMS receipt was received and updated in the system; it is used internally while tracking SMS receipts in between a particular time interval and user will get this time in latestStatusTime parameter.

g. Tracking Duplicate ctid Messages

If a message has been submitted with duplicate ctid and ctidunique is true in the settings then it will not be sent and while tracking status you will get the following response object.

```
{
    "gstid": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
    "gctid": "9c234op73b09b9",
    "statusCode": "001",
    "statusDescription": "Request Completed",
    "completionTime": "2015-09-23T18:04:24+0200",
    "receipts": [
        {
            "statusCode": "229",
            "statusDescription": "Duplicate ctid",
            "destination": "+27800000001",
            "ctid": "fdadsa-aadsfvsfdsad"
            "identifier": "test2",
"sentTime": "2017-02-20T09:08:39.000+02:00"

                    }
    ]
}
```

h. Tracking Invalid Messages

If a message has any invalid number then it will not be sent and while tracking status you will get the following response object.

Example message with Invalid number

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"250420160537",
"recipients":[
{"to":"27812300", "identifier":"123"}],
"messageText":"hello"
}
```

On tracking the above message you will get following response

Response Object

```
{
    "gstid": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
    "gctid": "9c234op73b09b9",
    "statusCode": "001",
    "statusDescription": "Request Completed",
    "completionTime": "2015-09-23T18:04:24+0200",
    "receipts": [
        {
            "statusCode": "226",
            "statusDescription": "Invalid Destination",
            "destination": "+27812300"
            "identifier": "test2"
```

```
                        }
    ]
}
```

i. Response for Archived Receipts

The client will get the following response if the receipt has been archived.

```
{
"receipts": null,
 "statusCode": "506",
"statusDescription":"ARCHIVED_AND_NOT_AVAILABLE",
"completionTime":"2016-04-21T21:22:26+05:30"
}
```

Delivery receipts will be archived after every 6 months. If the client tries to retrieve a 6 months old delivery receipt then it will not be available and the client will get the above message.

Delivery receipts can also be sent on provided the client's email ID. To activate getting replies on email client will have to choose send receipts on email option through online account.

j. Track SMS URL limit

Client can query for getting receipts by hitting the above provided URL's.
There is a limit for tracking status of SMS in a given time interval. A user can query for each ctid or stid ten times in the first 10 minutes interval, the first 10 minutes interval is counted from when the user had used tracking URL for the first time once he had exceeded that then he can query for each ctid or stid only once in every 10 minutes.
If a user tracks status of SMS using the track SMS by time option or by Repeat all receipts option then also he can track only 10 times in the first 10 minutes interval and later on he will be allowed to query once every ten minutes.

If client exceeds the tracking limit he will get the following error message.

```
{
"receipts": null,
"eSTId": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
"statusCode": "228",
"statusDescription":"TRACKING_LIMIT_EXCEEDED",
"completionTime":"2015-05-21T21:22:26+05:30"
}
```

15. Cancel Delivery / Delete Scheduled SMS

User can cancel sending scheduled SMS by deleting them before scheduled date has been reached.

To delete scheduled SMS you can use HTTP/S GET or POST as follows:

a. Delete Scheduled SMS Using HTTP GET

- **Delete Scheduled SMS by stid and destination number**

https://api.ecommunicate.co.za/sms/deleteScheduledSMS?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&stid=0a6a3de7-b974-42d0-af9e-9c23473b09b9&destination=27800000000

- **Delete Scheduled SMS By time**
  User can delete scheduled SMS in between a time period as follows:

https://api.ecommunicate.co.za/sms/deleteScheduledSMS?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&startdate=2016-04-22T00:45:26%2B02:00&enddate=2016-04-22T23:45:26%2B02:00

startdate: All messages scheduled after this date, including this date.

enddate: All messages scheduled before this date, including this date.

Note :Format of input dates must be YYYY-MM-DDTHH:mm:ssZ.

All dates will always be returned in YYYY-MM-DDTHH:mm:ssZ format to enable multiple time zones.

- **Delete All Scheduled SMS**

If user wants to delete the entire set of scheduled SMS for all the cell numbers that constitute the message then they can send ctid or stid along of that message.

https://api.ecommunicate.co.za/sms/deleteScheduledSMS?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&ctid=042220160337

or

https://api.ecommunicate.co.za/sms/deleteScheduledSMS?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&stid=eff47ce1-9a88-46b2-8fd8-1f9bedc2b4e1

To delete all Scheduled SMS you can use deleteAll parameter as follows

https://api.ecommunicate.co.za/sms/deleteScheduledSMS?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&deleteall=true

b. Delete Scheduled SMS Using HTTP POST

For deleting scheduled SMS using HTTP POST user must provide required parameters in JSON format as given below to **https://api.ecommunicate.co.za/sms/deleteScheduledSMS**:

- **Delete Scheduled SMS by stid and destination number**

  Send request to: **https://api.ecommunicate.co.za/sms/deleteScheduledSMS**

```
{
"stid":"0a6a3de7-b974-42d0-af9e-9c23473b09b9",
"destination":"27800000000",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

- **Delete Scheduled Sms By time**
  User can delete SMS scheduled in between a time period as follows:

  Send request to: **https://api.ecommunicate.co.za/sms/deleteScheduledSMS**

```
{
```

```
"startDate":"2016-04-22T00:45:26+02:00",
"endDate":"2016-04-22T23:45:26+02:00",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"

}
```

startDate: All messages scheduled after this date, including this date.

endDate: All messages scheduled before this date, including this date.

Note :Format of input dates must be YYYY-MM-DDTHH:mm:sZZ.

All dates will always be returned in YYYY-MM-DDTHH:mm:ssZ format to enable multiple time zones.

- **Delete All Scheduled Messages**

Send request to: **https://api.ecommunicate.co.za/sms/deleteScheduledSMS**

If user wants to get the entire set of scheduled SMS for all the cell numbers that constitute the message then they can send ctid or stid of that message.

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"ctid":"042220160337"
}
```

Or

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"stid":"0a6a3de7-b974-42d0-af9e-9c23473b09b9"

}
```

- **Delete Scheduled SMS By ctids**
  User can delete scheduled SMS using an array of ctid's used while sending SMS.

  Send request to: **https://api.ecommunicate.co.za/sms/deleteScheduledSMS**

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"ctids":["042220160337","042220160340"]
}
```

ctids: It will take array of ctid's .

- **Delete All Scheduled SMS**
  To delete all Scheduled SMS you can use deleteall parameter as follows
  Send request to: **https://api.ecommunicate.co.za/sms/deleteScheduledSMS**

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"deleteall":"true"

}
```

Response Object

If you have scheduled messages then you will get the following response

```
{
    "statusCode": "508",
    "statusDescription": "SCHEDULED_MESSAGES_DELETED",
    "completionTime": "2015-08-28T12:23:42+0200"
}
```

16. CallBackURL

Delivery reports are to acknowledge the client about sent SMS status. Replies are the messages sent to the client. These reports and replies will always be available on our website.
If client wants these to be sent on client's server then they can use callBackURL.

The client can provide callbackURL while registration process on online account or can provide it while sending JSON object via callBackURL parameter.

a. JSON format for sending SMS with callBackURL parameter

Whole format for sending SMS will be the same only one extra parameter must be added as follows in the JSON.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
"replyCallBackURL":"http://client.getstatus.co.za/sendSMS?......"
},
"to":"2790000000",
"messageText":"hello"
}
```

b. Receipt Call Back URL

Sample Call Back URL for receiving sent SMS status updates: This URL is an HTTP Get Request

http://client.getstatus.co.za/sendStatus?stid=7491d356-db10-465d-a4a2-e637f2a7947e&destination=%2B27111715198&smsID=d3f44eef-10cf-47d3-98e3-6e9713bbf9b0&ctid=Sat+Oct+20+10%3A42%3A00+SAST+2018&gstid=2a987d4e-ce7e-4eb1-9778-6dfbde6d9368&statusCode=500&statusDesc=Message+Sent.&deliveredtime=&senttime=2018-10-20T10%3A42%3A08.000%2B02%3A00

Used Parameters:

i. stid:Tracking ID of SMS provided in the response JSON after message is sent.
ii. destination:Destination mobile number on which SMS has been sent
iii. smsID : Unique ID for SMS returned by the gateway.
iv. ctid:it is the client tracking id, which should be provided with message request
v. gstid:Unique ID for SMS returned by the gateway.{Unique Id is the gstid we pass back the gstid with the receipt when sms is sent}
vi. statusCode:This represents the status code of the SMS
vii. statusDesc:This provides description of the status code.

viii. deliveredtime:Time at which message got delivered.
ix. senttime:Time at which message was sent by user.

c. Reply Call Back URL(Get)

Sample Call Back URL for receiving reply SMS :
This URL is an HTTP Get Request

http://messaging.ecommunicate.co.za/pl4sms/testReply?gstid=ba8a27aa-f737-4f5e-b218-0512c119b516&ctid=testingctid2&date=2018-08-10+15%3A21%3A33.571&from=%2B27737715199&message=Reply1

Used Parameters:

i. gstid: Unique ID for SMS returned by the gateway .{Unique Id is the gstid we pass back the gstid with the receipt when sms is sent}
ii. date: Date time of status update.
iii. from: Mobile number from which SMS has came.
iv. message: Text Message
v. ctid: it is the client tracking id, which should be provided with message request

If SMS status updates or replies has not delivered to client then above URL will be recalled after some intervals, this will continue till 24 hours from the first try of sending.

Callback URL will be called in the following intervals:

2 minutes ---- 4minutes ---- 8 minutes----16 minutes--- 32 minutes ---to ---720minutes --- 1440 minutes

- * Sometimes the status message of SMS may not be delivered to the client..
  So after sending SMS its status must be checked by the client for success or failure using TrackingMessageStatusURL provided in
-
- TRACKING MESSAGE STATUS.

d. Reply Call Back URL(Post)

Sample Call Back URL for receiving reply SMS :
This URL is an HTTP Get Request

http://client.getstatus.co.za/sendReply
Add Header for Authentication
Username:client_username

Password:password

Sample Request:

```
{
"time":"2016-04-22T23:45:26+02:00",
"from":"2790000000",
"reply":"hello",

}
```

If SMS status updates or replies has not delivered to client then above URL will be recalled after some intervals, this will continue till 24 hours from the first try of sending.

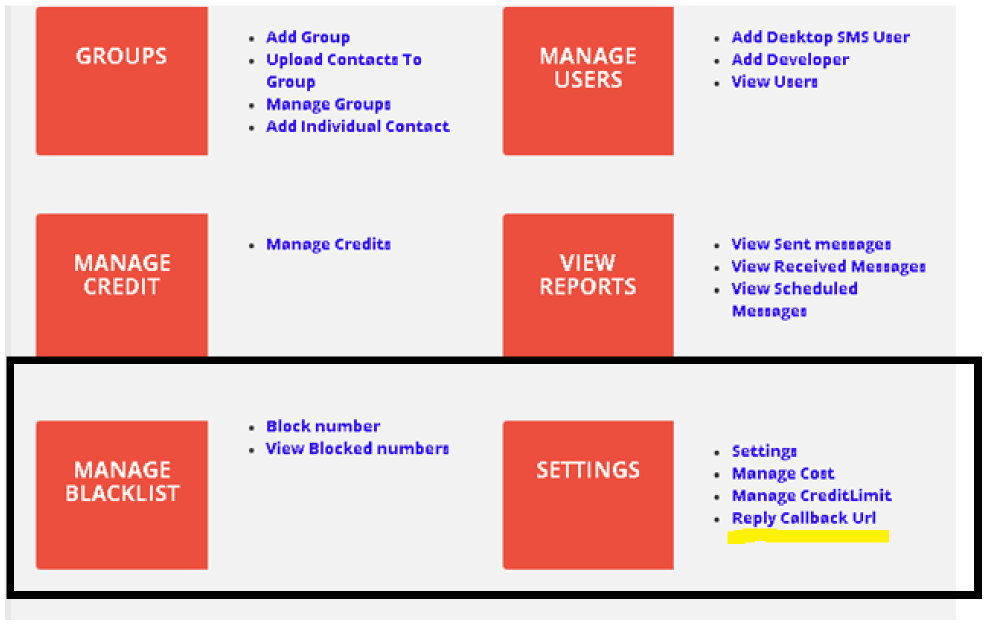
Callback URL will be called in the following intervals:

2 minutes ---- 4minutes ---- 8 minutes----16 minutes--- 32 minutes ---to ---720minutes ---1440 minutes

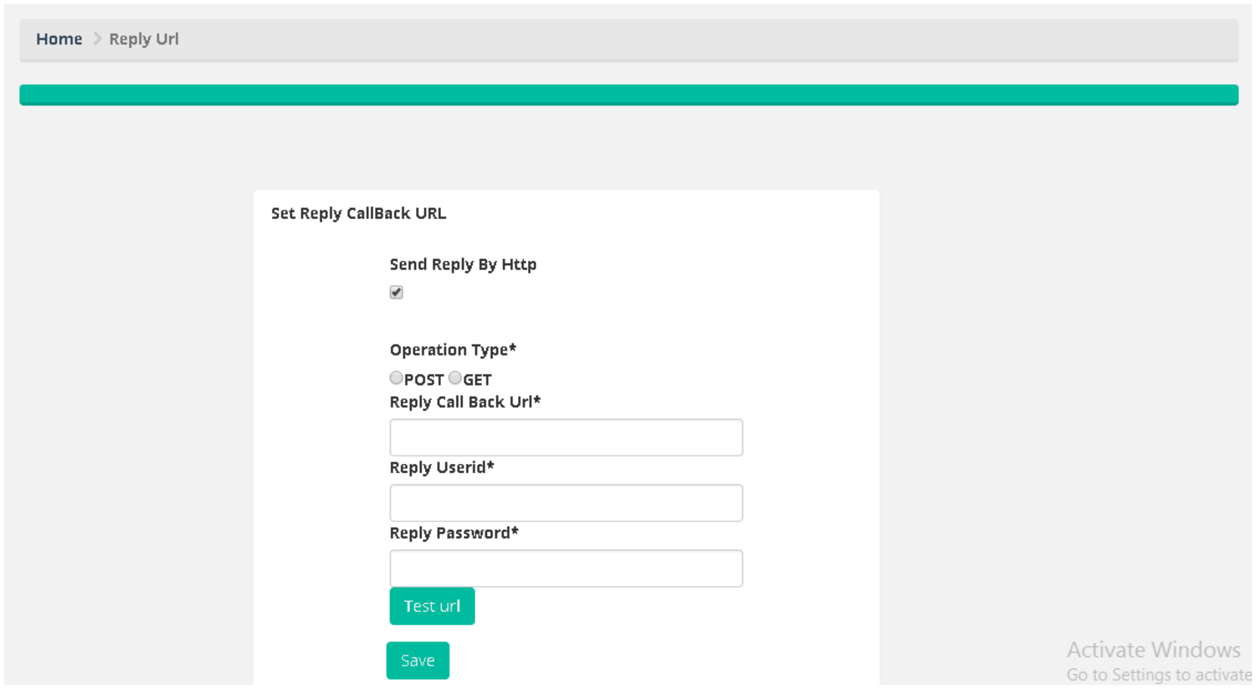
- * Sometimes the status message of SMS may not be delivered to the client..
  So after sending SMS its status must be checked by the client for success or failure using TrackingMessageStatusURL provided in
-
- TRACKING MESSAGE STATUS.

e. Set Reply Call Back URL On Website

Login to website in the home page you will see the following page click on the highlighted URL.



After clicking on the above link the following page will be displayed


Home > Reply Url

Set Reply CallBack URL

Send Reply By Http
☑

Operation Type*
○POST ○GET

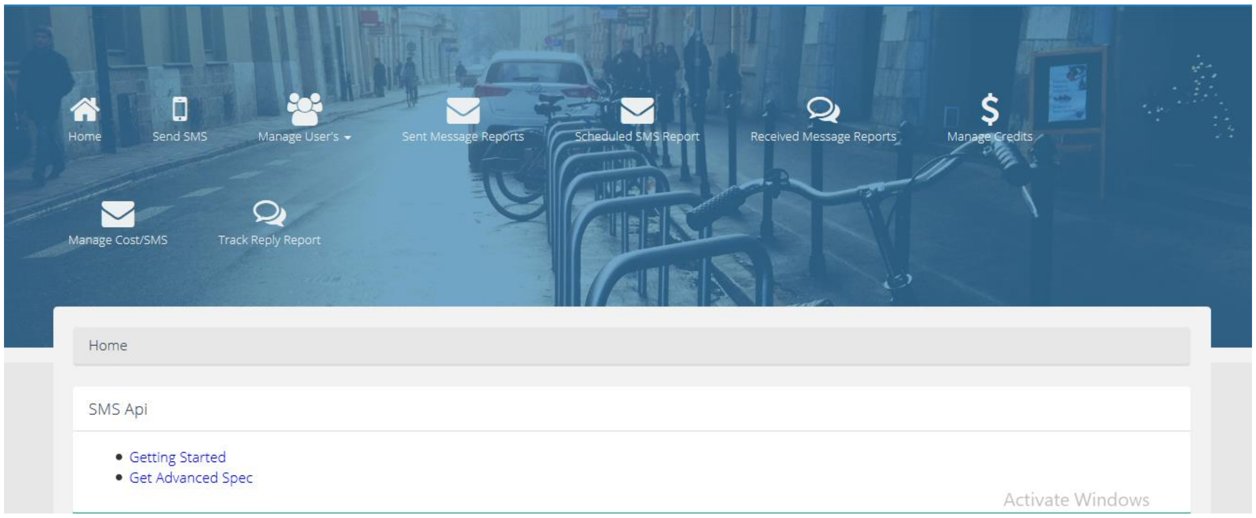
Reply Call Back Url*

Reply Userid*

Reply Password*

Add the Reply URL as per the request created (Get/Post) select accordingly, and add username and password on the form and after that we can even test the URL before adding by clicking **Test URL** button and then we can save it by clicking on **save** button.

f. Track Reply Call Back URL on Website

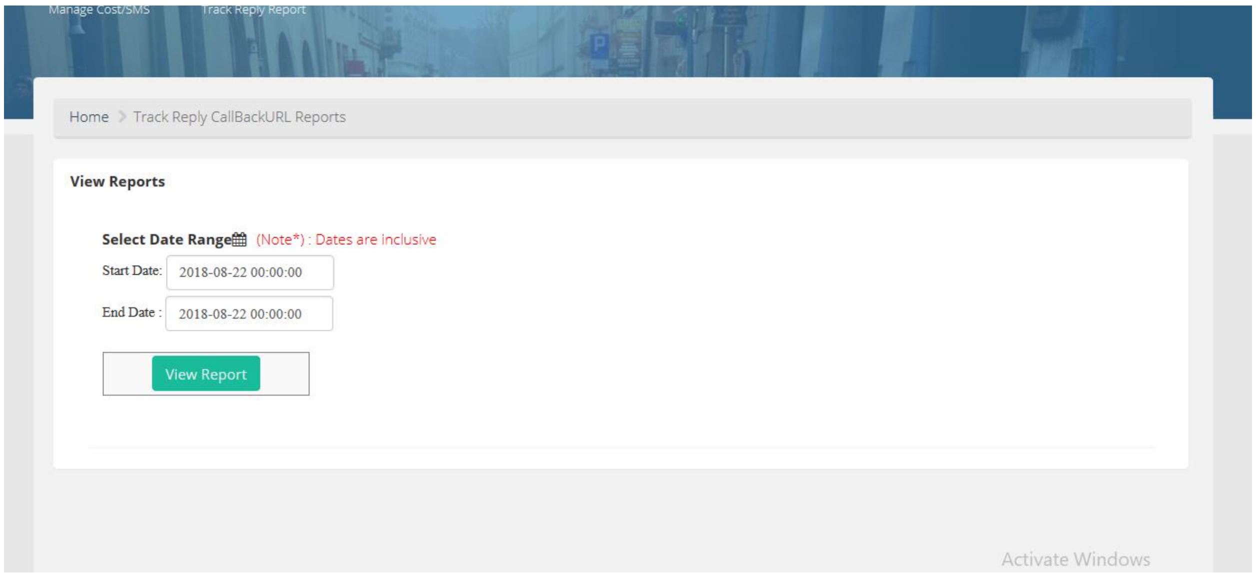


Home

SMS Api

- Getting Started
- Get Advanced Spec

On Menu Bar there is a Track Reply Reports Option, after clicking on it it will take us to the following page:



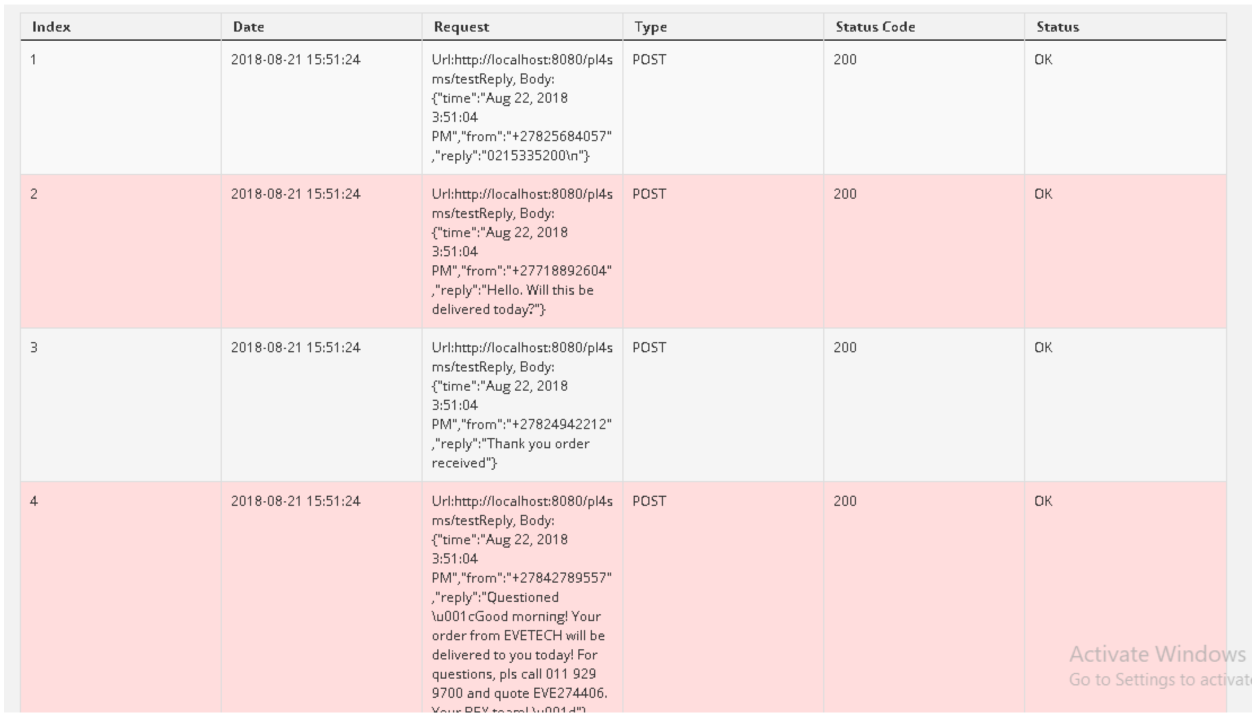
View Reports

Select Date Range (Note*) : Dates are inclusive

Start Date: 2018-08-22 00:00:00

End Date : 2018-08-22 00:00:00

here we need to select date for which we want to track the replies for and after that we need to click on view reports which will show the Track of replies sent to the respective Reply Call Back URL, the track reply report will look as the following image.



Index	Date	Request	Type	Status Code	Status
1	2018-08-21 15:51:24	Url:http://localhost:8080/pl4sms/testReply, Body: {"time":"Aug 22, 2018 3:51:04 PM","from":"+27825684057","reply":"0215335200\n"}	POST	200	OK
2	2018-08-21 15:51:24	Url:http://localhost:8080/pl4sms/testReply, Body: {"time":"Aug 22, 2018 3:51:04 PM","from":"+27718892604","reply":"Hello. Will this be delivered today?"}	POST	200	OK
3	2018-08-21 15:51:24	Url:http://localhost:8080/pl4sms/testReply, Body: {"time":"Aug 22, 2018 3:51:04 PM","from":"+27824942212","reply":"Thank you order received"}	POST	200	OK
4	2018-08-21 15:51:24	Url:http://localhost:8080/pl4sms/testReply, Body: {"time":"Aug 22, 2018 3:51:04 PM","from":"+27842789557","reply":"Questioned \u001cGood morning! Your order from EVETECH will be delivered to you today! For questions, pls call 011 929 9700 and quote EVE274406. Your REX team! \u001d"}	POST	200	OK

17. Dynamic Emails

To receive delivery and sent reports along with SMS replies on email address user can use dynamic Emails parameter as follows.

This parameter holds three inner parameters to allow different reply, delivery and sent report email ID. To provide more than one email ID for the below provided parameters separate them with comma (,) as given example.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{

"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"dynamicEmails":{
        "deliveryEmail":"testaccount2@test.com",
        "sendEmail":"testaccount2@test.com,test3.test.com",
        "replyEmail":"testaccount2@test.com"
}
},
"to":"27800000000",
"messageText":"hello"

}
```

If these parameters are not set then emails will not be sent for reports and replies.

18. Destination Address (to)

SMS messages can be sent in standard international format i.e. MSISDN format.

The term MSISDN (Mobile Station International Subscriber Directory Number) relates to a mobile number in international format with the "+" sign knocked off. In South Africa the MSISDN would be a cell number starting with 27.You can find further details via http://en.wikipedia.org/wiki/MSISDN. Our API can accept mobile numbers in three formats:

i. MSISDN Format: The mobile number should be in MSISDN format with the correct country code followed by the number. No leading zero to the number and no special characters such as "+" or spaces must be used.
   a. For example : 27800250005
ii. 9 Digit Mobile Number: In such case country code selected in client user account will be appended with that number the default selected country code is South Africa.
   a. For example :800250005
iii. 10 Digit Mobile Number (starting with zero): If user provides number starting with zero then zero will be removed and country code selected in client user account will be appended with that number the default selected country code is South Africa.

For example: 0800250005

The client can change default country code to any other from his online account.

a. Duplicate Destination:

If SMS message contains a destination address twice then duplicate SMS will be rejected and it will not be charged. To view all rejected Messages you can log on to the website.

b. Invalid Destination:

If SMS message contains invalid destination address that have incorrect format then those SMS will be rejected and not be charged. To view all rejected Messages you can log on to the website.

c. How to prevent duplicate SMSs

i) If you send single SMSs with the same cell number then we don't reject it.

Example below of Single messages sent with the same ctid and message and destination. By sending single SMS, the duplicate messages will not be rejected.

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"21042016042700",
"to":"27800000000",
"messageText":"hello simple post 1"

}

{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"21042016042700",
"to":"27800000000",
"messageText":"hello simple post 1"

}
```

However if in the settings section "ctidunique":true is used, then the second single message will not be sent and it is rejected as Duplicate ctid. The system will not allow the same ctid to be used for that day.

```
{
```

```
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"21042016042700",
"to":"27800000000",
"messageText":"hello simple post 1"

}

{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"21042016042700",
"to":"27800000000",
"messageText":"hello simple post 1"

}
```

ii) Example: Sending to multiple destinations and the message is rejected for duplicate destinations.

If you send multiple SMSs with the same cell number then we reject the second destination.

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27737715190", "identifier":"John"},
{"to":"27737715190", "identifier":"Anna"}],
"messageText":"hello123 "
},
{
"ctid":"210420160917",
"recipients": [
{"to":"27737715197", "identifier":"Lisa"},
{"to":"27737715196", "identifier":"Ron"}],
"messageText":"Friendly reminder that you can receive your account balance by SMSing
the keyword BAL followed by your 7 digit account number to 12547 and 7 digit account
number to 12349"
}

]

}
```

In all the requests you get back **Message Accepted for Processing** and in the details report on the website it says **Duplicate Destination**.
If we are sending a batch like above and have entered duplicate destinations then the messages will be rejected as **Duplicate Destination**.
In the above case the yellow marked Destination will be rejected as it has same destination as the above.

iii) If you set ctidunique to true, the system will not allow the same ctid to be used for that day - which means you will not be able to send a duplicate SMS by setting ctidunique to true and having a unique ctid per cell number.

Example: Duplicate destinations and Duplicate ctid.

```

"settings":{
"emailid":"testaccount2@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"ctidunique":true
},
"messages":[{
"ctid":"uweywiquyruietg",
"recipients": [
{"to":"27111892139", "identifier":"John"},
{"to":"27111892140", "identifier":"Anna"}],
"messageText":"duplicate check "
},
{
"ctid":"uweywiquyruietg",
"recipients": [
{"to":"27111892149", "identifier":"Lisa"},
{"to":"27111892169", "identifier":"Ron"}],
"messageText":"Friendly reminder that you can receive your account balance by SMSing"
}

]
}
```

If ctidunique is true and both batches of messages above have the same ctid then it will be rejected as Duplicate ctid.
In the above Example the 2nd batch will be rejected as it has same ctid as above.

Note* If ctidunique is true and if we are getting same ctid which is already sent for that day then that messages will also be rejected as Duplicate ctid

19. Message Rejected For Insufficient Credits

If you look at your attached report from our website it says INSUFFICIENT_CREDITS and that is why the SMSs were rejected.

If Messages are Rejected for INSUFFICIENT_CREDITS : We provide the following response

```
{
    "statusCode": "137",
    "statusDescription": "Insufficient Credit",
    "completionTime": "2017-12-06T10:05:01.000+02:00",
    "eSTId": "d0e1a9e5-1ace-4c33-b4bf-79b46bed412e"
}
```

20. Blacklisted

Blacklisting a number allows the subscriber to stop and block any incoming Message from a provider.

Customers can request not to receive any SMS from a particular client (sender of SMS) in such case number of that customer (recipient) will be added in a blacklist for that sender so that in future no messages will be sent to that recipient number until that has been removed from blacklist.

If a customer(recipient) sends a SMS reply with ‘END’, ‘CANCEL’, ‘UNSUBSCRIBE’ , ‘QUIT’ or ‘STOP’ text in it that means we should not send any further messages to them and thus we will add that recipient number into blacklist for the client for which the SMS reply did came.

If client tried to send SMS to a blacklisted number it would not be sent and client will get there credits back. Clients can view those SMS via user interface.

Also clients can block a number if they don’t want to send SMS to it .It can be done manually from the user interface or they can request by sending email to ecommunicate support to block a recipient number.

If a client wants to unblock a blacklisted number he should contact ecommunicate support for that.

21. MessageText

MessageText sent will be encoded in GSM 7 bit encoding format. Each messageText can have 160 characters and it will be charged for one messageText. All characters that are not in GSM 7 bit encoding format will be removed.

- Example:
  Hello 元元 Customer..

Send balance to 553 to get account information on your number. Text Offer to 55501 and get Exciting Offers for you. Dial 12101 to check account balance validity.

Above message have 元元 characters that are not allowed in GSM 7 bit encoding.

Those characters will be removed while sending messageText.

a. Concatenated Message:

When messageText length exceeds 160 characters then the messageText will split up to multiple separate SMS and will be sent separately. Each messageText then will have a special header called UDH (User Data Header) added at the beginning. This header states the order of each message.

A UDH takes 7 bits (6 octets) of a messageText thus rest of messageText can have up to 153 (134 octets) characters.

Thus each concatenated message can have each messageText with 153 characters in it.

- Example1:
  messageText **- Send balance to 553 to get account information on your number. Text Offer to 55501 and get Exciting Offers for you. Dial 12101 to check account balance validity.**

The above messageText contains 160 characters that make one message.

- Example2:

Hello customer.

Send balance to 5353 to get account information on your number. Text Offer to 535501 and get Exciting Offers. Dial

121 to check balance validity.

The above messageText contains 161 characters that make 2 messages with first messageText having 153 characters and second containing 8 characters.

To calculate length of SMS follow the below link:
https://messente.com/documentation/sms-length-calculator

http://messente.com/blog/2012/09/15/how-is-length-of-the-sms-calculated/

b. Special Characters:

The following are the only valid SMS characters that clients should use for non-Unicode SMSs. See **http://en.wikipedia.org/wiki/GSM_03.38**

The Basic SMS character set counting as 1 SMS character per character:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!\"#$%&'()*+,-./:;<=>?@_

Line feeds will be counted as 1 character.

This corresponds to the Default Alphabet for SMS. All other characters will be stripped out before sending the SMSs. You can send all characters using Unicode parameter. See Section Unicode Messages

22. Unicode Messages

The unicode system is able to handle large variety of characters beyond the basic Latin alphabet (A-Z), in order to support a wide range of languages and technical symbols.

Any message that contains even a single Unicode character must be encoded in this way in order to display the Unicode characters properly.

A normal SMS text message contains up to 160 characters from the GSM alphabet but a Unicode message can contain up to 70 characters before it must be split into multiple parts.

The following are the only valid SMS characters that clients should use for non-Unicode SMSs. See **http://en.wikipedia.org/wiki/GSM_03.38**

The Basic SMS character set counting as 1 SMS character per character:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!\"#$%&'()*+,-./:;<=>?@_

Line feeds will be counted as 1 character

All characters outside this character set can be sent via Unicode messages by setting Unicode parameter to true.

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount",
"clientRef":"testaccount"
},
"unicode":"true",
"ctid":"201606210422",
"to":"27737725199",
"messageText":"Estimado/a cliente, Saiba que o seguro anual ou semestral na GA
oferece serviço de reboque gratuito ao veículo segurado (oferta limitada a zona de
Luanda). Ter seguro é bom. Ter GA é óptimo."
}
```

If Unicode is set to true but message text does not contain any Unicode character then that message will be sent and charged as per normal SMS cost.
For example:

This message contains 171 characters so it will cost 2 credits (as per normal SMS cost).

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount",
"clientRef":"testaccount"
},
"unicode":"true",
"ctid":"201606210422",
"to":"27737725199",
"messageText":" Friendly reminder that you can receive your account balance by SMSing
the keyword BAL followed by your 7 digit account number to 12547 and 7 digit account
number to 12349"
}
```

If Unicode is set to true and message contains one or more Unicode characters it will be charged as per Unicode message cost.

For example:
This message contains Unicode characters, it have 190 characters and will be charged 3 credits.

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount",
"clientRef":"testaccount"
},
"unicode":"true",
"ctid":"201606210422",
"to":"27737725199",
"messageText":"Estimado/a cliente, Saiba que o seguro anual ou semestral na GA
oferece serviço de reboque gratuito ao veículo segurado (oferta limitada a zona de
Luanda). Ter seguro é bom. Ter GA é óptimo."
}
```

23. Virtual SIM CARD ID

Reply and delivery receipts to client will be sent on the client's source address or Virtual SIM card ID.

There will be a default ID provided by the system for all the clients. This will be a 16 characters long alphanumeric or numeric string depending on the country of a client.
If user wants a different senderID then the default then he must contact ecommunicate support.

User can have multiple Virtual SIM cards ID, in such case user must provide one Virtual ID in the from parameter that he wants to use while sending SMS. If this parameter is not there then default Virtual Id will be used.

24. Sending Time Parameters

Client can set todaysLatestSendTime, nextDayEarliestSendTime to control the delivery and sending time of SMS.

a. todaysLatestSendTime :

Todays latest send time is the time before which the SMS must be sent to the SMSC. This parameter takes date in format **YYYY-MM-DDTHH:mm:ssZ.**

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"todaysLatestSendTime":"2016-04-22T17:50:26+02:00"
},
"to":"27800000000",
"messageText":"hello"
}
```

This parameter takes current date with time, if user provides date other than current date then you will get the following response message.

```
{
  "statusCode": "028",
  "statusDescription": "Todays latest send time parameter cannot have future date ",
  "completionTime": "2016-04-21T14:05:26.000+02:00",
  "eSTId": "dd9e8701-05b0-4b92-aa22-5fe5a0c13337"
}
```

If latestSendTime is reached before the message can be routed to the external gateway then message won't be sent with the status code (Message Expired Before Sending) and credits will be given back. The user will get the following response object on tracking that SMS

Response Object:

```
{
    "statusCode": "001",
    "statusDescription": "Request Completed",
    "completionTime": "2016-10-21T17:58:10.000+02:00",
    "receipts": [
        {
            "stid": "ae383d4b-36f3-4524-98fa-de4d2b930731",
            "statusCode": "421",
            "statusDescription": "Message expired before sending by user request",
            "destination": "+27800000000"
        }
    ]
}
```

If user wants to continue sending the remaining message on the next day then he can use the nextDayEarliestSendTime parameter.

b. nextDayEarliestSendTime

This parameter takes date in format **YYYY-MM-DDTHH:mm:ssZ.**

This parameter is used to continue sending remaining messages on the next day if todayslatestsendtime has been reached on the present day.

For example: In the below example todays latest send time parameter is set to 08.00 PM on 22nd April this means messages will not be sent after 08.00 pm 22nd April. If messages are not sent before this time then the remaining messages will get scheduled on the next day earliest send time ie. 08.00 AM on 23rd April.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"todaysLatestSendTime":"2016-04-22T20:00:00+02:00",
"nextDayEarliestSendTime":"2016-04-23T08:00:26+02:00"
},
"to":"27800000000",
"messageText":"hello"
}
```

If this parameter is not set then all remaining messages will be failed and credits will be given back.

All scheduled messages can be tracked using trackingMessageStatus URL see: Tracking Scheduled SMS

Send request to: **https://api.ecommunicate.co.za/sms/trackingMessageStatus**

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"startdate":"2016-04-22 00:25:43",
"enddate":"2016-04-23 15:25:43",
"isScheduledSms":"true"
}
```

This parameter takes tomorrows' date with time, if user provides any other date then he will get the following response message

```
{
  "statusCode": "029",
  "statusDescription": "Next day earliest send time parameter can only have
tomorow's date ",
  "completionTime": "2016-04-22T14:09:15.000+02:00",
  "eSTId": "fbcc8d68-9e08-4218-91c9-14ab08d68244"
}
```

c. Scheduling SMS with Sending Time Parameters

Send request to: **https://api.ecommunicate.co.za/sms/json**

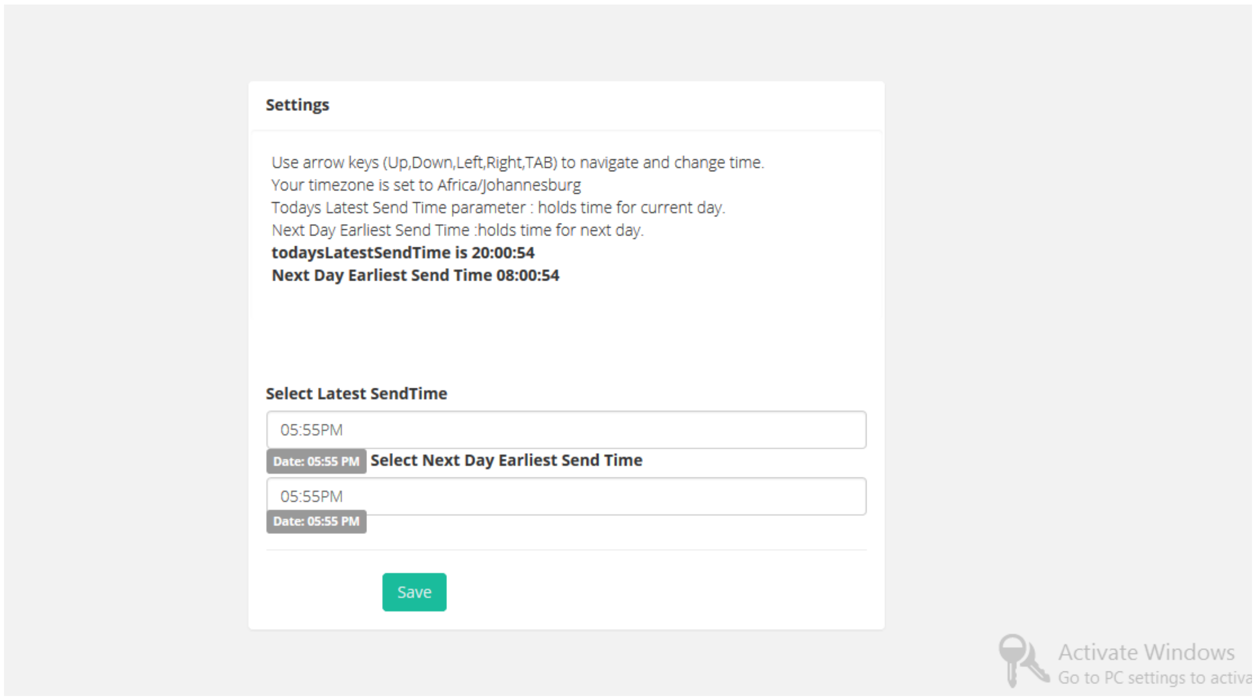
```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
 "scheduledSmsDate":"2016-04-13T17:50:26+02:00",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"nextDayEarliestSendTime":"2016-04-14T17:50:26+02:00"

},
"ctid":"abc1d2-37def",
"recipients": [
{"to":"2780000001", "identifier":"abcd"},
{"to":"2780000001", "identifier":"123"}],
"messageText":"hello whole batch scheduled "
}
```

While scheduling SMS todaysLatestSendTime parameter should have same date as scheduled date and nextDayEarliestSendTime should be next date of scheduled date otherwise it will give error.

d. Settings Section on Website:

User can set these two parameters (todayslatestSendTime and nextDayEarliestSendTime) from website also by going into Settings section.


Settings

Use arrow keys (Up,Down,Left,Right,TAB) to navigate and change time.
Your timezone is set to Africa/Johannesburg
Todays Latest Send Time parameter : holds time for current day.
Next Day Earliest Send Time :holds time for next day.
todaysLatestSendTime is 20:00:54
Next Day Earliest Send Time 08:00:54

Select Latest SendTime
05:55PM
Date: 05:55 PM
Select Next Day Earliest Send Time
05:55PM
Date: 05:55 PM

If these two parameters are set in website then they will be used as default values.

25. Message Parameters

a. Table of Parameters

NAME	PARAMETER	DESCRIPTION	DEFAULT VALUE	REQUIRED/OPTIONAL
Email Id	emailid	Email Id used while registration process		*Required*
Password	*password*	*Client password*		*Required*
clientref	*clientref*	Client Reference		*Required*

SessionId	sessionId	*ID for each sess*	*Optional*
Recipient Address	*to*	Recipients number	Required
From	from	Virtual SIM Card ID of user	Optional
Message Text	messageText	Text Message	*Required*
Identifier	identifier	Name of recipient	Optional
ClientTrackingID	ctid	Unique client id submitted by client with each message	*Optional*
SystemTrackingID	stid	System generated unique SMS message tracking Id	Required
ClientGlobalTrackingID	gctid	Unique client id submitted by client for all messages submitted in one requeset	Optional
Error System Tracking ID	eSTId	Unique Id to track error caused in sending messages	-
Latest Send Time	todaysLatestSendTime	Time	*Optional*
scheduledSmsDate	scheduledSmsDate	Date to schedule SMS to be sent on future date.	Optional
CallBack URL	receiptCallBackURL	URL on which	Optional

		receipts are sent	
	replyCallBackURL	URL on which replies are sent	Optional
Spec Version	specVersion	Version of the SMS SPEC API in use	Optional

26. Reply

This is a two way messaging service so the recipient can also send reply to the sender

Message replies sent by recipients to the client can always be viewed on website. If the client wants to receive replies on their server then they can use Reply Call Back URL.

Reply can also be sent on provided user email ID if user request for it. To activate getting replies on email User must choose send receipts on email option through online account.

a. Last Sent SMS Information

User will get the last sent SMS information with each reply. This will be a default case which is set by include sent SMS parameter set to true. If user doesn't want this information along with reply then he can use includeSentSms parameter as false.

b. Query Reply using HTTP GET

- **Query Reply By Time**

User can query for SMS receipts sent in between a time period as follows:

https://api.ecommunicate.co.za/sms/getReply?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&startDate=2016-04-24 18:25:43&endDate=2016-04-24 19:25:43&includeSentSms=true

startDate: Any reply received after this date, including this date.

endDate: Any reply received before this date, including this date.

*Format of dates must be same as given in example

c. Query Reply using HTTP POST

For tracking SMS using HTTP POST user must provide required parameters in JSON format as given below to **https://api.ecommunicate.co.za/sms/getReply**

- **Query Reply By Time**

```
{
"startdate":"2016-04-18 14:48:16",
"enddate":"2016-04-18 17:48:16",
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
 "includeSentSms":"true"
}
```

d. Response Object

The client will get back a response JSON as follows;

ResponseJSON:

```
{
   "replies": [
        {
           "number":"ABC",
           "identifier":"def",
           "messageText":"Hello testing",
           "replyReceived":"2015-05-21T21:22:26+05:30",
           "lastSentMessage":"Hello",
           "timeMessageSent":"2015-05-21T20:22:26+05:30"
          "ctid": "ccvae",
           "replyId": "1960522"
        }
],
   "statusCode":"001",
   "statusDescription":"REQUEST_COMPLETED",
   "completionTime":" 2015-05-21T21:22:26+05:30"

}
```

User will get the following response if the requested resource does not exist.

```
{
    "statusCode": "509",
    "statusDescription": "NO_REPLIES_FOUND",
    "completionTime":" 2015-05-21T21:22:26+05:30"
}
```

If includeSentSms parameter is false then response will be as follows:

```
{
"replies": [
        {
             "number": "ABC",
             "identifier":"def",
             "messageText": "Hello testing",
             "replyReceived": "2015-05-21T21:22:26+05:30"
             "timeMessageSent":"2015-05-21T20:22:26+05:30"
             "ctid": "ccvae",
             "replyId": "1960522"

        }
],
    "statusCode": "001",
    "statusDescription": "REQUEST_COMPLETED",
    "completionTime":" 2015-05-21T21:22:26+05:30"

}
```

e. Track Reply URL limit

There is a limit on using the track reply URL's .Each user can track reply by the tracking URL only 10 times in the each ten minutes interval. If this limit exceeds then user will get the following response. In such case you have to wait for 10 minutes and try again.

```
    {
 "eSTId": "0a6a3de7-b974-42d0-af9e-9c23473b09b9",
"statusCode": "228",
"statusDescription":"TRACKING_LIMIT_EXCEEDED",
"completionTime":"2015-05-21T21:22:26+05:30"
}
```

f. Reply Email:

Reply email will contain parameters as shown below:

Subject: SMS Reply from Cell No: 27813.....

Number: 27813.....

Reply Message Id: 0a6a3de7-b974-42d0-af9e-9c23473b09b9

Message: Yes

Time Received: 2015-03-30 08:51:46.005

Last Message Sent: testing email reply

Time Sent: 2015-03-30 08:51:06.41

Used Parameters:

i. Number : Number of user who has sent reply message.
ii. replyMessageId : System generated unique Id for reply message.
iii. Message: Reply message content
iv. Time Received : Reply message receiving time
v. Last Message Sent : Last message sent the that user
vi. Time Sent : Message Sending Time

27. Credit Balance

It’s important to know there are enough credits before sending SMS to prevent error code 029.

Client will get emails about credit balance in their account there will be one email that will be sent on daily basis regarding the credits and other email will be sent on reaching a limit which can be set by user via online account.

Credit Balance email will contain the following parameters:

Subject: SMS Account Balance

Emailid :testClient

Current Balance : 20000

To view the credit balance of user account, User can use query credit URL or the user interface.

a. Get Balance using HTTP GET

To use HTTP/S get for querying user must hit the below provided URL along with his emailid, password ,clientref.

https://api.ecommunicate.co.za/sms/getBalance?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2

Get Balance using HTTP POST

To use HTTP POST for querying user must provide a JSON object with required parameters to

https://api.ecommunicate.co.za/sms/getBalance

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

b. Response Object

The response object will contain information about the credit balance along with one more parameter called testGateway , if it's true that means it's a test account and messages will not be sent in real.
If it's a real account then that parameter will not be shown.

```
{
    "statusCode": "001",
    "statusDescription": "Request Completed",
    "emailid": "testaccount2@test.com ",
    "currentBalance": "210",
    "testGateway":"true"
}
```

To query balance from User Interface user must login to his online account with his credentials.

27. Get Unsubscribed Destinations

To get the Unsubscribed number within custom date , the following api can be used https://api.ecommunicate.co.za/sms/getUnsubscribelist

Request Json:

```
{
"startDate":"2017-12-01",
"endDate":"2017-12-08",
"emailid":"testaccount2@test.com",
"password":"Testaccount2",
"clientref":"testaccount2"
}
```

Response Object:

```
{
    "unsubscribers": [
        {
            "number": "27737715199"
        }
    ],
    "status": "Blacklisted List"
}
```

28. Server Time

Clients that are sending delayed messages would like to know what the current time on the server is.

The time returned is in **YYYY-MM-DDTHH:mm:ssZ** format.

a. Get Server Time using HTTP GET

To use HTTP get for querying user must hit the below provided URL along with his emailid, password ,clientref set servertime parameter to true.

https://api.ecommunicate.co.za/sms/getServerTime?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2

b. Get Server Time using HTTP POST

To use HTTP POST for querying user must provide a JSON object with required parameters to **https://api.ecommunicate.co.za/sms/getServerTime**

```
{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
}
```

c. Response Object

```
{
    "statusCode": "001",
    "statusDescription": "Request Completed",
    "emailid": "testaccount2@test.com ",
    "serverTime": "2015-10-20T13:01:25.000+02:00"
}
```

29.Failure Recovery

A system failure can occur because of a hardware failure or a severe software issue; causing the system to freeze, reboot, or stop functioning altogether. This may happen sometime in middle of sending message and user may not get response back.

In such case user will have to resend the whole JSON again along with the same ctid.

For example
User sent a JSON object like this

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"250420160639",
"to":"27800000000",
"messageText":"hello"
}
```

Some failure occurred in between and user did not get response object.

Then user must resend the previous JSON with same ctid.
For example:

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"250420160639",
"to":"27800000000",
"messageText":"hello"
}
```

In such case user will receive accepted response as follows:

1) Response Object

```


{

"eSTId": "null",
"statusCode": "100",
"statusDescription": "MESSAGE_ACCEPTED_FOR_PROCESSING",
"completionTime":"2015-05-21T21:22:26+05:30",
"acceptedMessageCount": "1",

}
```

But in the backend if the message was sent already then it will not be sent again and credits will be given back to client and unsent message can be viewed via website or Tracking URLs.

Note: To recover from failure and to avoid sending already sent SMS, user must provide unique ctid while sending SMS. If ctid is not present in a message then that cannot be tracked in case of failure and will be sent again.

30.Specversion, Commsversion, Appversion

If this document is used to define the code for an application the specversion keyword should be set to 1.0. To enable companies to integrate this spec into their environments these three parameters are provided for backward compatibility and support. If these keywords are not used the SMS will still be sent, but they are useful to provide backward compatibility and so we request for them to be used, especially the specversion keyword.

The commsversion is used where the communication layer can change such as by using a DLL. By programming the commsversion i.e. the version of the DLL it's possible to view what technology was used to send the SMSs and so easily update the DLL where previous versions had problems. Furthermore various applications can be used to send SMSs and should there be a problem with a customer that can't send SMSs.

The appversion can be used to view the version and type of the application sending SMSs. This is useful to find out wether message was sent via Desktop application or via Website.
User can view all Sent SMS with all information via UI and user can find out that SMS was sent via desktop or website.

31.Examples

This part is having two types of examples provided.

i. Simple: With only basic required parameters.
ii. Advanced: With all parameters.
iii. SessionId: Example of sending message using sessionId parameter.

a. Sending Priority Message for Banking with DSM ON : -

For sending text message to single destination with SMS not saved in our database and with priority just provide the below URL with required parameters.

For example:
If your emailid is : testaccount@test.com then you will have to url encode " @" and "." as follows.

**https://api.ecommunicate.co.za/sms/json?emailid=testaccount%40test%2Ecom&password=
test123&clientref=test&to=0278300000&messageText=Your%20otp%20is%208888&dsm=true&priority=true.**

Used Parameters:

b. emailid : email Id used while registration.

c. password : password
d. clientref : clientref
e. to: mobile number of receiver. See Destination Address (to)
f. messageText : Text to be sent
g. dsm : Set dsm as true to not save message on our server.
h. priority: Set priority to true or false, when priority is set true the sms is transactional, and will be sent via premium route, for example priority can be set true for banking OTP(One Time Password). When priority is set false the sms is promotional, and will be sent via regular route, for example advertisement messages. Default is false.

Note: When priority is set true the SMS may cost more
Note: Priority flag should be string with value true or false.

b. Simple Message Examples:

Simple Example of sending one message to one recipient.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"to":"27800000000",
"messageText":"hello"
}
```

Advanced Example of sending one message to one recipient .

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
"specVersion":"1.0"
},
"ctid":"250420160639",
"to":"27800000000",
"messageText":"hello"
}
```

Example of sending one message to one recipient using sessionId.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"sessionId":"0a6a3de7-b974-42d0-af9e-9c23473b09b9",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
"specVersion":"1.0"
},
"ctid":"250420160639",
"to":"27800000000",
"messageText":"hello"
}
```

Simple example of sending one message to multiple recipients.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```

{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"250420160638",
"recipients":[
{"to":"27800000000", "identifier":"john"},
{"to":"27800000000", "identifier":"lia"}],
"messageText":"hello"
}
```

Advanced Example of sending one message to multiple recipients .

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
```

```
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
"specVersion":"1.0"
},
"ctid":"250420160640",
"recipients":[
{"to":"27800000000", "identifier":"john"},
{"to":"27800000000", "identifier":"lia"}],
"messageText":"hello"
}
```

Example of sending one message to multiple recipients' usingsessionId.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"sessionId":"0a6a3de7-b974-42d0-af9e-9c23473b09b9",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
"specVersion":"1.0"
},
"ctid":"250420160641",
"recipients":[
{"to":"27800000000", "identifier":"john"},
{"to":"27800000001", "identifier":"lia"}],
"messageText":"hello"
}
```

c. Messages Examples

Simple example of sending messages.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"gctid":"250420160650"
},

"messages":[{
"ctid":"250420160642",
"recipients":[
{"to":"27800000001", "identifier":"john"},
{"to":"27800000000", "identifier":"jenny"}],
"messageText":"hello people"

},
{
"ctid":"250420160643",
"recipients":[
{"to":"27800000002", "identifier":"lia"},
{"to":"27800000003", "identifier":"ron"}],

"messageText":"hello 1234"
}
]
}
```

(I) Advanced example of sending messages.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"gctid":"250420160651",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
 "specVersion":"1.0"

},
"messages":[{
"ctid":"250420160643",
"recipients":[
{"to":"27800000003", "identifier":"nit"},
{"to":"27800000001", "identifier":"sia"}],
"messageText":"hello"
},
{
"ctid":"sewasvxasew",
"recipients":[
{"to":"27800000002", "identifier":"ron"},
{"to":"27800000004", "identifier":"amy"}],

"messageText":"hello 1234"

}
]


}
```

(II) Advanced example of sending messages.

Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2",
"gctid":"250420160654",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
"replyCallBackURL":"http://client.getstatus.co.za/sendSMS?......",
"specVersion":"1.0"
},
"messages":[{
"ctid":"250420160645",
"recipients":[
{"to":"27800000004", "identifier":"joe"},
{"to":"27800000003", "identifier":"john"}],
"messageText":"hello"
},
{
"ctid":"250420160646",
"recipients":[
{"to":"27800000005", "identifier":"ria"},
{"to":"27800000002", "identifier":"tom"}],
"messageText":"hello"
}
]
}
```

(III) **(Example of sending messages using sessionId.**
Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"sessionId":"0a6a3de7-b974-42d0-af9e-9c23473b09b9",
"gctid":"250420160657",
"todaysLatestSendTime":"2016-04-13T17:50:26+02:00",
"receiptCallBackURL":"http://client.getstatus.co.za/sendStatus?......",
"replyCallBackURL":"http://client.getstatus.co.za/sendSMS?......",
"specVersion":"1.0"
},
"messages":[{
"ctid":"250420160648",
"recipients":[
{"to":"27800000004", "identifier":"james"},
{"to":"27800000003", "identifier":"joe"}],
"messageText":"hello"

},
{
"ctid":"250420160649",
"recipients":[
{"to":"27800000002", "identifier":"tia"},
{"to":"27800000001", "identifier":"ria"}],
"messageText":"hello"
}
]
}
```

d. Other Examples

A message may have invalid destination number then in such case that message will not be sent.
Send request to: **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"ctid":"250420160701",
"recipients":[
{"to":"27800000000", "identifier":"joe"},
{"to":"27812300", "identifier":"amy"}],
"messageText":"hello"
}
```

Response Object:
User will get the following response on submitting the Message.

```
{
"ctid": "abc123def",
"stid": "eff47ce1-9a88-46b2-8fd8-1f9bedc2b4e1",
"statusCode": "100",
"statusDescription": " MESSAGE_ACCEPTED_FOR_PROCESSING ",
"completionTime":"2016-04-21T21:22:26+05:30"

}
```

User will get information about invalid destination message while tracking SMS status.

https://api.ecommunicate.co.za/sms/trackingMessageStatus?emailid=testaccount%40test%2Ecom&password=testaccount2&clientref=testaccount2&stid=eff47ce1-9a88-46b2-8fd8-1f9bedc2b4e1&sentdate=2016-04-21

Response Object:

```
{
    "statusCode": "001",
    "statusDescription": "REQUEST_COMPLETED",
    "completionTime": "2016-04-21T22:06:14.000+05:30",
    "receipts": [
        {
            "stid": " eff47ce1-9a88-46b2-8fd8-1f9bedc2b4e1",
            "statusCode": "400",
            "statusDescription": "QUEUED_AT_MNO_FOR_ACCEPTANCE",
            "destination": "+27800000000",
             "identifier":"joe "
        },
{
            "statusCode": "226",
            "statusDescription": "Invalid Destination",
            "destination": "+27812300",
            "ctid": " abc123def",
            "identifier":"amy"

        }

    ]
}
```

32. Appendix 1: Validation Error Messages

SMS goes through a validation process before getting sent. If an SMS fails validation process of the system then user will get the following error codes and description as response. Messages will not be charge in case of these errors.

STATUS CODE	DESCRIPTION	DETAIL
001	Request Completed	Completed the request.
021	Authentication Failed	Authentication details are incorrect
022	Invalid Parameters	One or more required parameters are invalid
023	Missing Parameters	One or more required parameters are missing
024	SessionId Expired	Your session ID is expired
025	SessionId Not Exist	Session Id does not exist. Please request a new session Id .

026	Simultaneous Request Limit Reached	User cannot send more simultaneous messages as maximum limit is reached. The limit for simultaneous request is set to 2 by default. If you want to increase the simultaneous request limit please email support@ecommunicate.co.za .
027	Duplicate Global ctid	The gctid parameter value is duplicated.
028	Today's Latest Send Time Parameter Cannot Have Future Date	Todayslatestsendtime parameter takes current date for sending normal SMS.
029	Next Day Earliest Send Time Parameter Can Only Have Tomorrow's Date	nextdayearliestsendtime parameter takes tommorow's date for sending normal SMS.
030	Today's Latest Send Time Parameter Should Have Same Date As Scheduled Date	Todayslatestsendtime parameter takes current date for sending scheduled SMS.
031	Next Day Earliest Send Time Parameter Can Only Have Next Date Of Scheduled Date	nextdayearliestsendtime parameter next date of scheduled date for sending scheduled SMS.
032	Scheduled Time Must Be Greater Than Current Time	Messages can be scheduled only to future date.

33.Appendix 2: Status Messages

Status messages are generated once an SMS passes validation process. Messages that are failed after validation in this sending process can be checked on the website.

ERROR CODE	ERROR DESCRIPTION	DETAILS
100	Message(s) Accepted For Processing	Message(s) is accepted for processing.
101	Message(s) Not Processed	Message(s) not yet processed.

126	Blacklisted Destination	This number is not allowed to receive messages.
128	Invalid Destination	Destination number is invalid
129	Duplicate Destination	Duplicate Message
130	Tracking Limit Exceeded	The client has used trackingMessageStatusURL more than once in 10 minutes. Please try later.
131	Expired Before Sending	Message expired before sending by user request
134	Duplicate ctId	clientTrackingID given by user already exists
135	Null Message	Message text is empty
136	Message Batch Size Exceeded	Message is having more than 1000 recipients.
137	Insufficient Credits	You do not have enough credits to send messages.
139	Credits are less than total batch messages cost	You do not have sufficient credits to send that batch message.
140	Empty Recipient	Recipient is empty
200	Loaded For Sending	Message has been processed successfully and loaded for sending
201	Message Batch Scheduled	Messages has been scheduled successfully
400	Accepted By Internal SMS Gateway	Message has been accepted by our SMS gateway and will now be sent to the mobile networks which will send it to the recipient.

480	Message(s) Accepted For Processing	If the OTP Token is invalid the status code is set to **480 and the SMSs will then be sent via the default queue** on the primary SMS gateway and not by the high priority OTP queue.
500	Message Sent.	Message received by gateway
501	Queued At MNO For Delivery	Confirmed delivered to gateway or the mobile network
502	Message(s) Accepted For Processing	For those clients that require 24/7 uptime and who use the backup gateway: **If the status code is set to 502 please send the SMSs via the backup gateway.**
503	Stale	Message sent. The maximum time to receive a delivery receipt is now over and no further delivery receipt will be received.
504	No Reference To Batch	Messages does not exist
505	Message Not Sent	Message related to that query was not sent.
506	Archived	Receipt is archived and thus not available.
507	Scheduled Messages Not Found	Scheduled Messages does not exist.
508	Scheduled Messages Deleted	Scheduled Messages deleted Successfully
509	No Replies Found	No reply messages have found
520	Rejected By Mobile Network	General category including all the reasons the message is rejected by the mobile network and not sent.
600	Delivery Success	Delivered to Recipient
620	Failed In Delivery	The SMS failed to be delivered

34. Terminology

- SMS : Multimedia Messaging Service.
- SMSC:Short Message Service Center.
- MSISDN: Mobile Station International Subscriber Directory Number.
- URL: Uniform Resource Locator
- emailid: Name of the external content provider. The client may have a few emailid's and passwords all using the same clientref for different system calling the SMS Gateway. This enables the client to differentiate between the different systems that are calling the SMS gateway.
- password: Authorized password for the user.
- clientref : clientRef – This is the unique reference for the client.
- to : Phone number of recipient along with proper country code
- ctid : TrackingID provided by the client while sending the SMS request. This is an optional parameter.
- messageExpiry: This field sets the expiry date of SMS.SMS allows at most 5 days of expiry period. User can set the value between 1-5 days.SMS will be expired after the days set in this field and will not be delivered.
- statusCode: Status code of Sent SMS.
- statusDescripition : Description of status of sent SMS.
- errorCode : Error codes that are sent to user when SMS gateway rejects the SMS.
- errorDescription: Description of error codes sent to the user when SMS gateway rejects the SMS.
- REST :Representational State Transfer (REST) view http://en.wikipedia.org/wiki/Representational_state_transfer

35. Testing Json :

Please use credentials of testing account Provided to you:

Sending Multiple Message:

Send request to : **https://api.ecommunicate.co.za/sms/json**

```
{
"settings":{
"emailid":"testaccount@test.com",
"password":"testaccount2",
"clientref":"testaccount2"
},
"messages":[{
"ctid":"210420160916",
"recipients": [
{"to":"27111892139", "identifier":"John"},
{"to":"27111892140", "identifier":"Anna"}],
"messageText":"hello123 "
},
{
"ctid":"210420160917",
"recipients": [
{"to":"27111892149", "identifier":"Lisa"},
{"to":"27111892169", "identifier":"Ron"}],
"messageText":"Friendly reminder that you can receive your account "
}

]
}
```

Note * The Recipient Number Strictly should begin with 27111 otherwise the messages will be rejected as Invalid Destination.
For Example:

```
{
"settings":{
"emailid":"testaccount2@test.com",
"password":"Testaccount2",
"clientref":"testaccount2"
},
"messages":[{
"ctid":"364dhsfgdshd",
"recipients": [
{"to":"27112892140", "identifier":"John"},
{"to":"27111892140", "identifier":"Anna"}],
"messageText":"hello123 "
},
{
"ctid":"364dhsfgdshd",
"recipients": [
{"to":"27111892149", "identifier":"Lisa"},
{"to":"27111892169", "identifier":"Ron"}],
"messageText":"Friendly reminder that you can receive your account balance by SMSing"
}

]

}
```

For the above example:

```
{"to":"27112892140", "identifier":"John"},
```

Has msisdn starting which do not start with 27111 , so it will be rejected as Invalid Destination

You can track the information with our tracking url and will get the following response

```
{
{
    "gstid": "57ba9c57-7f37-44c1-8b1b-364755778f90",
    "statusCode": "001",
    "statusDescription": "Request Completed",
    "completionTime": "2017-08-03T13:57:01.000+02:00",
    "receipts": [
        {
            "id": 0,
            "stid": "de90b262-a698-4389-a9dc-5c9153e69a81",
            "statusCode": "128",
            "statusDescription": "Invalid Destination",
            "destination": "27112892140",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:06:35.000+02:00",
            "messageText": "hello123 ",
            "identifier": "John",
            "msgcost": "1"
        },
        {
            "id": 0,
            "stid": "de90b262-a698-4389-a9dc-5c9153e69a81",
            "statusCode": "600",
            "statusDescription": "Delivered to Recipient",
            "destination": "+27111892140",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:06:47.000+02:00",
            "sentTime": "2017-08-03T13:06:36.000+02:00",
            "messageText": "hello123 ",
            "identifier": "Anna",
            "msgcost": "1"
        },
        {
            "id": 0,
            "stid": "fc5df4f8-02f4-4b02-af59-26584731cc4f",
            "statusCode": "600",
            "statusDescription": "Delivered to Recipient",
            "destination": "+27111892149",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:06:47.000+02:00",
            "sentTime": "2017-08-03T13:06:36.000+02:00",
            "messageText": "Friendly reminder that you can receive your account
balance by SMSing",
            "identifier": "Lisa",
            "msgcost": "1"
        },
        {
            "id": 0,
            "stid": "fc5df4f8-02f4-4b02-af59-26584731cc4f",
            "statusCode": "600",
            "statusDescription": "Delivered to Recipient",
            "destination": "+27111892169",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:06:47.000+02:00",
            "sentTime": "2017-08-03T13:06:36.000+02:00",
            "messageText": "Friendly reminder that you can receive your account
balance by SMSing",
            "identifier": "Ron",
            "msgcost": "1"
        },
        {
            "id": 0,
            "stid": "d8edbc17-0e95-42bc-b096-0255e68b8fbc",
            "statusCode": "128",
            "statusDescription": "Invalid Destination",
            "destination": "27112892140",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:45:35.000+02:00",
            "messageText": "hello123 ",
            "identifier": "John",
            "msgcost": "1"
        },
        {
            "id": 0,
            "stid": "d8edbc17-0e95-42bc-b096-0255e68b8fbc",
            "statusCode": "600",
            "statusDescription": "Delivered to Recipient",
            "destination": "+27111892140",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:45:47.000+02:00",
            "sentTime": "2017-08-03T13:45:36.000+02:00",
            "messageText": "hello123 ",
            "identifier": "Anna",
            "msgcost": "1"
        },
        {
            "id": 0,
            "stid": "4bd89887-4747-46e7-b47c-98db86e457dc",
            "statusCode": "600",
            "statusDescription": "Delivered to Recipient",
            "destination": "+27111892149",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:45:47.000+02:00",
            "sentTime": "2017-08-03T13:45:36.000+02:00",
            "messageText": "Friendly reminder that you can receive your account
balance by SMSing",
            "identifier": "Lisa",
            "msgcost": "1"
        },
        {
            "id": 0,
            "stid": "4bd89887-4747-46e7-b47c-98db86e457dc",
            "statusCode": "600",
            "statusDescription": "Delivered to Recipient",
            "destination": "+27111892169",
            "ctid": "364dhsfgdshd",
            "deliveredTime": "2017-08-03T13:45:47.000+02:00",
            "sentTime": "2017-08-03T13:45:36.000+02:00",
            "messageText": "Friendly reminder that you can receive your account
balance by SMSing",
            "identifier": "Ron",
            "msgcost": "1"
        }
    ]
}}
```

37 Future Implementations:

Some of the points given in the specifications have not been implemented yet. Those are as follows:

i. Virtual SIM CARD ID
ii. Message Validity Period
iii. Follow Marketing Rules
iv. CallBackUrl