Your Partner In Your Mobile & Online Presence

Last Invention® SMS Platform Developer API
User Manual

lastinvention.co.za

CONTENTS

1. Version Control

Revision Number	Revision Date	Author	Changes
0.1	30 Sep 2010	Thato Dipudi	Initial Document Release
0.2	05 Mar 2017	Thato Dipudi	SMS platform URL update
0.3	04 May 2017	Thato Dipudi	Updated document name and trademark information
0.4	22 Apr 2018	Thato Dipudi	Added section: Retrieval of your project's Last Invention platform information

2. SMS Developers Guide - Introduction

This document is intended to serve as a guide to how 3rd party client applications need to integrate with the Last Invention® SMS Platform and also serves as a guideline in the development phase.

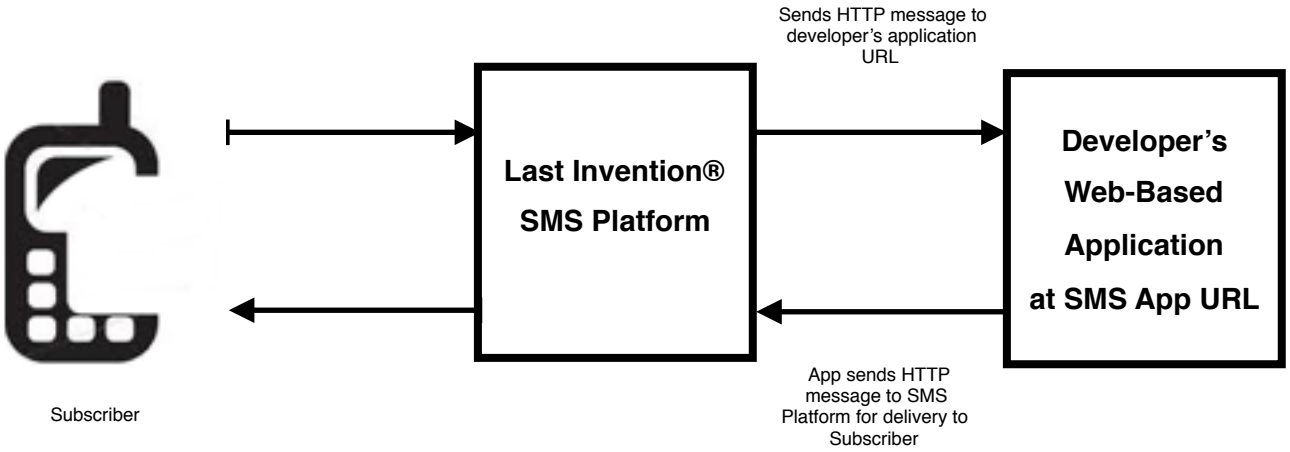

Figure 1.1: Process Flow

3. Client Requirements

The developer is required to have adequate infrastructure to communicate to the Last Invention® SMS Platform gateway and host 3rd party applications. This includes internet connectivity, compatible web servers and/or application platforms, etc. The implementation centres around *URL based requests* and *responses*.

The developer would also need to signup for the Last Invention® SMS Platform service (free service, only pay for SMS credits).
For more information about Last Invention® SMS Platform, please goto the Our Website. Also note that you SMS credits need to be purchased as well.

After signing up for the Last Invention® SMS Platform, the following will be available:

a) GUI credentials (username and password): for you to send SMSs from the online GUI. These will be sent to you via email.
b) API credentials (username and password): these will be the details you will need for the API integration (different to the GUI credentials). You will find these by logging into the GUI and navigating to: *Service* >> *Manage Service*, then on the *General Config* tab, look on the *API Credentials* section.

4. Provisioning

For more information about Last Invention® SMS Platform go to the Our Website.

The following will be needed as your API credentials:

- username : as per section 3 (b)
- password : as per section 3 (b)
- account name : same as username

The developer must supply the following application specific information so that SMS Platform can send messages to your application:

- MOSMS (mobile device originating SMS that is sent to as a reply to the SMS sent from your SMS platform) Endpoint URL, including port, at which your application listens for incoming message.
- Delivery Report (DLR) Endpoint URL, including port, at which your application listens for incoming status / delivery reports.

The developer can update the MOSMS URL and DLR URL as follows:

- For MOSMS URL: Log into the GUI and navigating to: *Service >> Manage Service*, then on the *General Config* tab. Change *HTTP MOSMS Endpoint* to your app URL.
- For DLR URL: Log into the GUI and navigating to: *Service >> Manage Service*, then on the *General Config* tab. Change *HTTP DLR Endpoint* to your app URL.

5. Short Message Service (SMS) Messages

Message content handling:
Messages are plain text. The only formatting options are the \n that will force a new line. This new line is included in the character count along with any other non printing characters.
All characters are supported in SMS. The characters that are included in the SMS do however affect the maximum characters that you can include in a message.

Long message characters:
To enable the long message parts to be re-combined a header is included within each SMS message which takes seven 8-bit characters. Hence a long message over 160 characters (7-bit) will cause the first message to be reduced in length by 7 characters and these will be sent in the second message. So for example, if a single long message of 161 characters (7-bit) is sent then this is actually transmitted and billed as two SMS messages; the first containing the first 153 characters, and the second, the remaining 8 characters. This is extended to the third, fourth and fifth message parts, as follows:

Number of SMS	Maximum usable characters (7-bit)
1	160
2	306
3	459
4	612
5	765

Threading messages:
Most modern mobile operating systems group messages using the source ID assuming that they come from the same sender/contact. This is typically called threading and can be controlled using the API with the threadid parameter. The thread ID alters the source address in an effort to separate messages into separate threads on the handset. This can give the user the illusion that the messages are unrelated and will allow you to track responses to individual unrelated messages. The user id that you get for a response will only relate to the last message in a thread, Typically this is the required behaviour as the user would reply to messages sequentially in the thread. If you don't include the threadid then all messages to a msisdn will appear in the same thread.

Available delivery mechanisms:
The API's can be used using http GET or POST. GET methods are easy to test but can leave your messages open to interception. Transparent proxies can store your SMS data in their log files and the urls are easy to sniff. GET requests can be used to send to 200 users in one request. POST requests are harder to debug but the requests are less likely to get intercepted. You can send up to 10000 messages concurrently with the parameter POST method and the XML POST allows you to send to an 'unlimited' amount of MSISDNs (Mobile Station International Subscriber Directory Number).

6. Mobile Terminated SMS (MT)

SMS messages submitted to gateway by the developer app to be delivered to a GSM subscriber, is referred to as a Mobile Terminated SMS (MT).

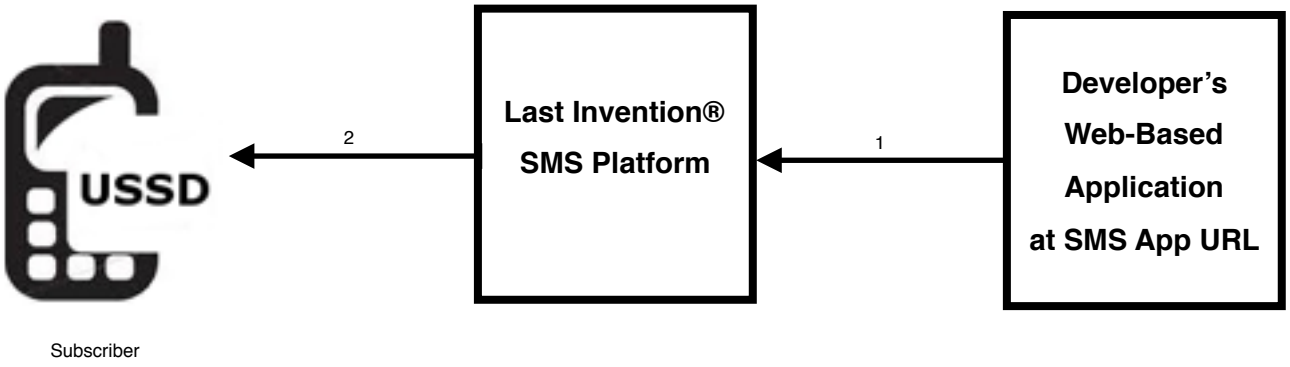

Figure 6.1: MTSMS data flow

Parameter description:

Name	Type	Description
username	String	Username as per section 3 (b)
password	String	Password as per section 3 (b)
account	String	Account as per section 3 (b)
da	String	Destination address for SMS (msisdn) this must always include the country code and can optionally start with a +. When using the bulk interface this parameter can contain a comma separated list of msisdns.
ud	String	Message to be sent in the SMS.
id	String	Optional unique identifier for SMSs this can be used to map MTSMSs to delivery reports.
schedule	Timestamp format YYYY-MMDD HH:MM:SS	Optional GET parameter for the Bulk XML upload only. Facilitates a scheduled send.
batchid	String	This option sets the batch ID as viewed in the campaign manager.
threadid	int	Optional threadId parameter which controls which thread a message goes into on the handset. Due to the way handsets thread messages you will only receive responses to the last message in a thread.
tag	deprecated	Provides the same functionality as threadId

6.1. Single SMS

Single shot SMSs should be sent using http GET or POST requests and should be sent to the following URL.

Base URL: http://mysms.lastinvention.co.za
Single SMS URL: http://mysms.lastinvention.co.za/submit/single/

SMSs sent to this interface are aggregated into a single batch on the Last Invention® SMS Platform GUI.

Authentication:

The client application must include the provided username, password and account name in the HTTP request URL. All fields must be properly URL encoded. These parameters are included in the GET or POST parameters.

Response codes:

The following error codes are generated on this interface.

HTTP/1.1 401 Unauthorized
"Invalid account, username, and/or password"
HTTP/1.1 400 Bad Request
"..."
HTTP/1.1 500 Internal Sever Error
"..."
HTTP/1.1 202 Accepted
"'Accepted for delivery\n<smsid>"

HTTP GET Example

Parameters

?username=SSS&password=SSS&account=SSS&da=SSSSSS&ud=SSSSS&&id=SSS

Sample Request

GET /submit/single/?username=bob&password=19jdasfs&account=test&da=
%2b8273829387&ud=Hello+World&id=msg01 HTTP/1.1
Host: mysms.lastinvention.co.za

Sample Response

HTTP/1.1 202 Accepted
Content-Length: 123
Content-Type: text/plain
Accept

HTTP POST Example

Parameters

username=SSS&password=SSS&account=SSS&da=SSSSSS&ud=SSSSS&id=SSS
Content-Type
application/x-www-form-urlencoded

Sample Request

POST /submit/single/ HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 105
username=foo&password=bar&account=smpp.example&da=%2b27102345789&id=123&ud=Hello+World

Sample Response

HTTP/1.1 202 Accepted
Content-Length: 123
Content-Type: text/plain
Accepted for delivery\n<smsid>

6.2. Bulk SMS

When sending multiple SMSs at a time send to this interface.

Base URL: http://mysms.lastinvention.co.za
Single SMS URL: http://mysms.lastinvention.co.za/submit/batch/

This interface accepts GET, parameter (form) POST and XML POST.
Batches are generated that can be viewed in the campaign manager. **Please don't use this for single shot SMSs as it will make reporting unmanageable.**
The client is required to set the following HTTP header(s) when performing the HTTP request:

- Content-Type: Specifies the content type of the HTTP message. The required values are application/xml or text/xml for XML POST or application/x-www-formurlencoded for parameter POST.
- Host: Specifies the host for which this message is intended.

Authentication:

The client application must include the provided user name, password and account name in the HTTP request URL. All fields must be properly URL encoded. These parameters are included in the GET parameters.

XML Request Format

XML messages submitted by the client to the gateway for sending have the following format:

```
<MessageBatch>
    <Message da=”SSS” id=”SSS”>SSS</Message>
    <Message da=”SSS” id=”SSS”>SSS</Message>
    <Message da=”SSS” id=”SSS”>SSS</Message>
    ...
</MessageBatch>
```

MessageBatch
Each *MessageBatch* element must contain one or more *Message* element(s).

Message
Each *Message* element must contain a *da* attribute, and the message content.

- Destination Address (*da*)
  The destination MSISDN in international format.
  Example: +2782000000001

- Message Identifier (*id*)
  Alphanumeric message identifier, set by client, that can be used to identify a specific message. The message identifier will accompany associated Delivery Report message(s), unmodified.
  For more information, please see the *Delivery Reports (DLR) (section 8)*.
  Example: abc123

XML Response Format
When submitting a message, the gateway will immediately return an initial Status / Error message, in the following format:

```
<MessageBatch>
    <BatchIdentifier>SSS</BatchIdentifier>
    <Error>
        <Code>NN</Code>
        <Description>SSS</Description>
        <Info>SSS</Info>
    </Error>
</MessageBatch>
```

MessageBatch
The *MessageBatch* element contains one *BatchIdentifier*, and one *Error* element.

Error
The *Error* element contains one *Code*, *Description* and *Info* element. The *Error* element is used to determine whether the batch was successfully submitted. The *Description* element provides additional information related to the value contained within the Code element.
The Info element contains the number of SMSs that were successfully accepted.

Error Codes
The following error codes are generated on this interface. I've entered the logical groups below. Individual errors have explicit descriptions to aid with problem solving.

6xx Authentication Error
7xx Accepted
8xx Invalid or Bad Request
9xx Gateway Error

XML POST Example

Sample Request

POST /submit/batch/?username=bob&password=19jdasfs&account=test&batchid=batch01
HTTP/1.1
Host: mysms.lastinvention.co.za
Content-type: application/xml

```
<MessageBatch xmlns="http://xml.netbeans.org/schema/ExternalMessageView">
    <Message da=”+27822938432” id=”msg01”>Hello World!</Message>
    <Message da=”+27828273728” id=”msg02”>Hello World!</Message>
    <Message da=”+27829483829” id=”msg03”>Hello World!</Message>
</MessageBatch>
```

Sample Response

HTTP/1.1 200 OK
Content-Length: 236
Content-Type: text/xml

```
<MessageBatch xmlns="http://xml.netbeans.org/schema/ExternalMessageView">
    <BatchIdentifier>batch01</BatchIdentifier>
    <Error>
        <Code>700</Code>
        <Text>Accepted</Text>
        <Info>3</Info>
    </Error>
</MessageBatch>
```

HTTP GET Example

Parameters

?username=SSS&password=SSS&account=SSS&da=SSSSSS&ud=SSSSS&id=SSS

Sample Request

GET /submit/batch/?username=bob&password=19jdasfs&account=test&da=
%2b8273829387,278123456781&ud=Hello+World&id=msg01&batchid=batch01 HTTP/1.1
Host: mysms.lastinvention.co.za

Sample Response

BatchIdentifier: batch01
Code: 700
Description: Accepted
Info: 2

HTTP POST Example

Parameters

username=SSS&password=SSS&account=SSS&da=SSSSSS&ud=SSSSS&id=SSS

Content-Type

application/x-www-form-urlencoded

Sample Request

POST /submit/batch/ HTTP/1.1
Host: mysms.lastinvention.co.za
Content-Type: application/x-www-form-urlencoded
Content-Length: 105
username=foo&password=bar&account=smpp.example&da=
%2b27102345789,27831234568&id=123&ud=Hello+World

Sample Response

HTTP/1.1 202 Accepted
BatchIdentifier:
Code: 700
Description: Accepted
Info: 2

7. Mobile Originating SMS (MO)

SMS messages sent by a GSM subscriber, to be delivered to the client application by the gateway, is referred to as a Mobile Originated SMS (MO).

This section is relevant to
- SMS replies to all outgoing SMSs
- SMSs sent to Connect short and long codes

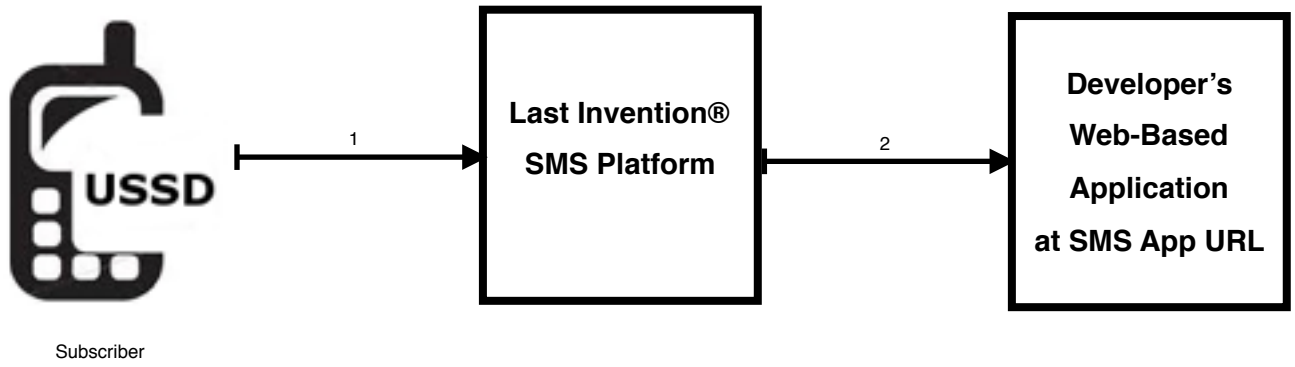

Figure 7.1: MOSMS data flow

Parameter description:

Name	Type	Description
ud	String	User Data.The text sent by the sender.
oa	String	Originating Address. The msisdn of the sender. The format includes the country code e.g. 27XXXXXXXXX
userid	String	The userid used when sending the original SMS.This field is used to map the response to the request.
charset	String	Identifies the charset that was used to send the SMS. Possible values are UTF-8 or UTF-16. You will use UTF-8.
timestamp	Timestamp format YYYY-MMDD HH:MM:SS	The timestamp indicating the time we received the SMS from the networks.
sms_id	int	An internal identifier. This is only likely to be of use to you if you have a query.
batchid	String	The batch id that was set in the message submission.
da	String	The destination msisdn that the message was sent to. This is only really useful when receiving MO SMS' to short codes.
tag	Deprecated	See userid
unix_epoch	Deprecated	Ignore

Example
Messages are delivered (individually) to the client's MOSMS Endpoint URL using HTTP GET requests.

Requests have following format:

?sms_id=1234&oa=27123456789&ud=Hello+World&timestamp=[yyyy-mm-dd hh:mm:ss]&charset=UTF-8&userid=123a

8. Delivery Reports (DLR)

After the client has submitted a MT SMS, and has been accepted for delivery, the gateway will report on the delivery status using one or more Delivery Report Messages.
Each Delivery Report Message will contain the 3rd party alphanumeric identifier provided by the client. The identifier is set using the id attribute inside the Message element. For more information, please see the Mobile Terminated SMS (MT)
The availability of delivery report is not guaranteed, and is subject to various factors, including network availability and congestion, subscriber availability, sufficient storage on the subscriber's handset, etc. Deliver reports are often delayed in peak network conditions.

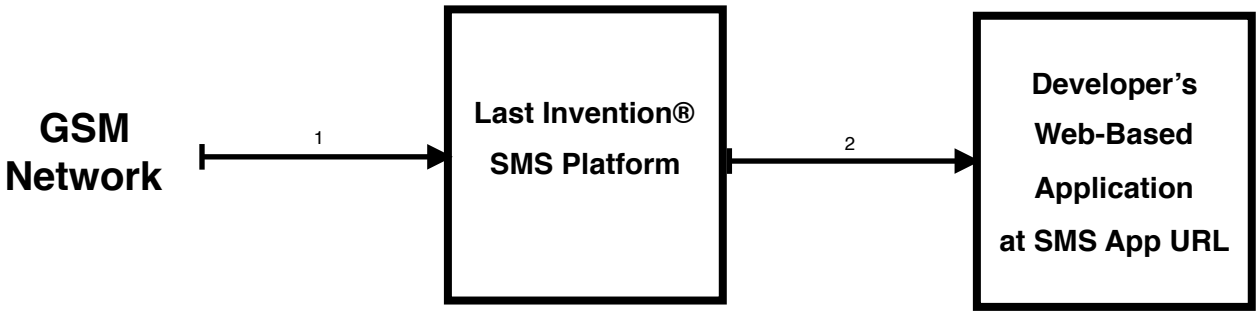

Figure 8.1: Delivery reports data flow

Parameter description:

Name	Type	Description
dlr	Short Int	The value of the delivery report. The composition of this is below.
da	String	The destination address (msisdn) that the sms was originally sent to. This is the full msisdn starting with the country code.
smsid	String	This relates to the response received when submitting the SMS via the single shot interface. See userid for batch interface.
submitted	Timestamp	YYYY-MM-DD HH:MM:SS
userid	String	This contains the id that was (optionally) with the SMS. This can be used to match DLRs to MTSMS.
batchid	String	The batch id that was set in the message submission.
oa	Deprecated	Ignore

Requests
Delivery Reports are delivered (individually) to the client's Delivery Report Endpoint URL using HTTP GET requests.

Delivery Report HTTP GET requests have following format:

?dlr=[bitmasked delivery status]&smsid=[internal gateway id]&userid=[specified id]

Note: The *userid* parameter will only be included in the request if the *id* attribute was set within the *Message* element when the batch was submitted for sending. For more information, please see the *Mobile Terminated SMS (MT)*.

Delivery Report Status:

The *delivery report status* is bit-masked, therefore, one or more delivery reports may be combined into a single message.

Bit-mask	Status	Description
1	DELIVERED	The SMS has been delivered to the subscriber.
2	FAILED	Delivery of the SMS has failed.
4	QUEUED	The SMS has been queued for delivery.
8	SENT	The SMS has been sent, but not yet delivered.
16	REJECTED	The SMS has been rejected by the SMSC.
32	ACCEPTED	The SMS has been accepted by the gateway.
64	REJECTED2	The SMS has been rejected by the gateway.

Example

```
GET /dlr/?dlr=41&smsid=12345&clientid=msg01 HTTP/1.1
Host: dlr developer app url
```

The delivery status of 41 can be interpreted as follows:

DELIVERED: ((41 & 1) = 1) ? TRUE
FAILED: ((41 & 2) = 2) ? FALSE
QUEUED: ((41 & 4) = 4) ? FALSE
SENT: ((41 & 8) = 8) ? TRUE
REJECTED: ((41 & 16) = 16) ? FALSE
ACCEPTED: ((41 & 32) = 32) ? TRUE
REJECTED2: ((41 & 64) = 64) ? FALSE

Alternatively:

DELIVERED: ((41 | 1) = 41) ? TRUE
FAILED: ((41 | 2) = 41) ? FALSE
QUEUED: ((41 | 4) = 41) ? FALSE
SENT: ((41 | 8) = 41) ? TRUE
REJECTED: ((41 | 16) = 41) ? FALSE
ACCEPTED: ((41 | 32) = 41) ? TRUE
REJECTED2: ((41 | 64) = 41) ? FALSE

9. Retrieval of your project's Last Invention platform information

You can retrieve your project's Last Invention platform account information via the following API.

Retrievable information:

a. Your company name (as stored on the platform) => entity_company_name.
b. Your unique project identifier => entity_project_id.
c. The package of the service your project is using => entity_package.
d. The number of USSD application menus => ussd_no_of_menus. Note that for Developer Program services, this tag will have a value of 0.
e. The cell phone number of the project administrator => admin_number.
f. The email address of the project administrator => admin_email_address.
g. The project status (ACTIVE / INACTIVE) => account_status.
h. The number of SMS credits in the account => sms_credits_balance. For the developer program, we keep it at 5 because we may have system generated SMS alerts that the account admin needs to receive (example is when the account reaches their low-threshold of 100 session credits).
i. The number of USSD session credits in the account => ussd_sessions_credits_balance.
j. The date and time that the project was created on the Last Invention platform => date_created.

A. You need to call the URL ...

http://ussd.lastinvention.co.za/ussd/clientaccount/?ussd_code=xxx&proj_id=yyy
Where…
xxx = URL encoded USSD code, and
yyy = your project Id
You need to request your project Id from the support department. Please keep the project Id secure.

B. The response from our server will be …

```
<?xml version="1.0" encoding="utf-8" ?>
<ussd_acc_info>
   <entity_company_name>Your company name on our system</entity_company_name>
   <entity_project_id>Your project id</entity_project_id>
   <entity_package>MyOwn_SME_USSD</entity_package>
   <ussd_no_of_menus>8</USSD_no_of_menus>
   <admin_number>27824638004</admin_number>
   <admin_email_address>name@domain.co.za</admin_email_address>
   <account_status>ACTIVE</account_status>
   <sms_credits_balance>1000.0</sms_credits_balance>
   <ussd_sessions_credits_balance>572.0</ussd_sessions_credits_balance>
   <date_created>2019-08-16 14:02:01.786330</date_created>
</ussd_acc_info>
```

If there is no account for the enquiry, the response will be …

```
<?xml version="1.0" encoding="utf-8" ?>
<ussd_acc_info>
   <entity_company_name></entity_company_name>
   <entity_project_id></entity_project_id>
   <entity_package></entity_package>
   <ussd_no_of_menus></USSD_no_of_menus>
   <admin_number></admin_number>
   <admin_email_address></admin_email_address>
   <account_status>Account does not exist</account_status>
   <sms_credits_balance></sms_credits_balance>
   <ussd_sessions_credits_balance></ussd_sessions_credits_balance>
   <date_created></date_created>
</ussd_acc_info>
```

NB:
The xml structure may be amended later. Check the latest API document for any changes.

10.Support

We've tried to put all the information we could in this Guide, but we are sure you'll have some questions.

If you would like to chat to us, please contact on any of the following channels:

- Phone: +27 (0) 82 463 8004
- WhatsApp: +27 (0) 82 463 8004
- Fax: +27 (0) 86 547 5309
- USSD Client Platform (South Africa only): Dial *120*5533#
- Skype name: lastinvention

Email:

- Sales: sales@lastinvention.co.za
- Support: support@lastinvention.co.za
- General Info: info@lastinvention.co.za

Business Hours:

- Time Zone: GMT + 2 (Pretoria, South Africa)
- Monday - Friday: 09H00 to 17H00
- Saturday: 09H00 to 12H00
- Closed on Sunday and Public holidays (Holiday Schedule)

Please also visit our website and submit a support ticket at:

Secure Client Portal: https://my.lastinvention.co.za/

Main website: https://lastinvention.co.za/

SMS platform webpage: https://lastinvention.co.za/sms-platform/

11.Trademark notice

Last Invention and **MyCVMate** are registered trademarks of The Last Invention (Pty) Ltd.