



Introduction

All Lyric VoIP and SMS devices include a web API for sending SMS remotely and handling the received messages. The device integrates a web server which handles HTTP GET queries, where messages to deliver are specified. The syntax for the queries is described below. All answers from server are in JSON format [1], which contains the result of the query sent.

JSON is a lightweight text-based open standard data interchange. Its structure is a collection of name/value pairs, for example:

```
{ "name": "John", "last_name": "Doe", "age": 22 }
```

Most high level programming languages include libraries to easily handle JSON objects. Third party libraries are available for practically every programming language.

The device holds messages in different circular queues for delivery and reception, each one with a capacity of 1000 elements. There is one queue for delivery for each channel, one general queue for delivery where messages would be sent by any channel and one queue for reception.

An example application written in Python [2] that reads the SMS message information (destination and content) from a CSV (comma-separated values) file, sends the messages through available gsm channels and waits for each one to finish with either success or fail status, can be found at the Yx support web site [3].

This document describes version 0.05 of the Web API.

Changelog

v0.05

- Fixed function *api_reset_sent_counter* not setting counters to 0 permanently
- Added the following functions:
 - *api_rcv_number_unread*
 - *api_rcv_get_first_n_unread*
 - *api_rcv_get_last_n_unread*
 - *api_sms_delete_by_status*
 - *api_sms_delete_by_id*
 - *api_sms_available_space*
- Added optional parameter *set_read* to functions *api_rcv_get_first_n* and *api_rcv_get_last_n*

v0.04

- Added function *api_get_channels_status*
- Added function *api_reset_sent_counter*

v0.03

- Added optional parameter *channel* to function *api_queue_sms*
- Added optional parameter *channel* to function *api_get_status*

- Added the following functions:
 - `api_sms_send_disable`
 - `api_sms_send_enable`

Web API Commands

Most of the following commands require authentication with the web server. The parameters used for authentication are:

username: Username for SMS API service authentication
 password: Password for SMS API service authentication
 api_version: Web API version used by the client

The `api_version` specified must match the one supported by the firmware version installed on the device. The examples encountered through the documentation use “web_user” as *username* and “web_pass” as *password*.

The details of each command will indicate if it requires the authentication parameters or not along its own parameters.

api_queue_sms:

Description: Queues the text message indicated in the query parameters and returns a ticket associated with the SMS. This ticket can be used later to ask for the delivery status of the message

Parameters:

content:	Content of the test message in URL format
destination:	SMS destination number
channel (optional):	Channel that should send the message (1 to 6). If not specified, the message will be sent by any enabled channel

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success:	1: Message was successfully inserted in the device database 0: An error occurred
message_id:	Ticket of the queued message. If an error occurred, this field won't be present on the response
error_code:	Only if an error occurred. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_queue_sms&username=lyric_api&password=lyric_api&content=SMS+Content&de
stination=55555555&api_version=0.05&channel=2

Answer:
{ "success": true, "message_id": 5 }

api_get_status:

Description: Retrieves the status of the message associated with the ticket indicated in the parameters

Parameters:

message_id:	Ticket of the message to be queried
channel (optional):	Get status for message associated with this channel. If not specified, the status returned will be from messages belonging to the general queue

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success:	1: Message status was retrieved successfully 0: An error occurred
message_status:	If an error occurred, this field won't be present on the response. 0: New 1: Processing 2: Sent 3: Failure
error_code:	See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_get_status&username=lyric_api&password=lyric_api&message_id=5&api_version
=0.05&channel=1

Answer:
{ "success": true, "message_status": 0 }

api_get_channels_status:

Description: Retrieves the status (a part of it on this version) of all the available channel

Parameters: No additional parameters required

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success: 1: Status of the channels retrieved successfully
0: An error occurred

error_code: Only if an error occurred. See error codes section

channels: A JSON array with the following info for each element (channel):

- id: Numeric channel id (from 1 to 6)
- sms_send_ena: 1 if sms delivery is enabled, 0 if not
- state: String indicating the channel status
- imsi: If state != "disabled" and state != "no_simcard", IMSI of the simcard
- n_sent_sms: Same conditions as imsi. N° SMSs sent by the simcard

Example: Query:
`http://web_user:web_pass@192.168.200.10/cgi-bin/exec?cmd=api_get_channels_status&username=lyric_api&password=lyric_api&api_version=0.05`

Answer:
{ "success": true, "channels": [{ "id": 1, "sms_send_ena": 1, "state": "registered", "imsi": "7300115555555", "n_sent_sms": 5 }, { "id": 2, "sms_send_ena": 0, "state": "registered", "imsi": "7300116666666", "n_sent_sms": 0 }] }

api_reset_sent_counter:

Description: Resets the SMS sent counter for the simcard of the IMSI or channel specified

Parameters:

- imsi: IMSI of the target simcard, or
- channel: Channel where the target simcard is inserted

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success: 1: Counter was reseted successfully
0: An error occurred

error_code: Only if an error occurred. See error codes section

Example: Query:
`http://web_user:web_pass@192.168.200.10/cgi-bin/exec?cmd=api_reset_sent_counter&username=lyric_api&password=lyric_api&api_version=0.05&channel=3`

Answer:
{ "success": true }

api_get_version:

Description: Returns the Web API version supported by the device

Parameters: No additional parameters required

Auth: No

Returns: The result (in JSON format) of the function contains the following element:
api_version: Web API version supported by the device

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?cmd=api_get_version

Answer:
{ "api_version": "0.05" }

api_get_number_recv_sms:

Description: Obtains the number of received messages

Parameters: No additional parameters required

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:
success: 1: Number of received messages obtained successfully
0: An error occurred
n_sms: If no error occurred, this field will be present on the response containing the number of received messages
error_code: Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?cmd=api_get_number_recv_sms&username=lyric_api&password=lyric_api&api_version=0.05

Answer:
{ "success": true, "n_sms": 500 }

api_recv_get_number_unread:

Description: Obtains the number of unread received messages

Parameters: No additional parameters required

Auth: Yes

Returns:	The result (in JSON format) of the function contains the following elements:
success:	1: Number of unread received messages obtained successfully 0: An error occurred
n_unread:	If no error occurred, this field will be present on the response containing the number of unread received messages
error_code:	Present when success field is 0. See error codes section

Example:	<p>Query:</p> <pre>http://web_user:web_pass@192.168.200.10/cgi-bin/exec?cmd=api_rcv_get_number_unread&username=lyric_api&password=lyric_api&api_version=0.05</pre> <p>Answer:</p> <pre>{ "success": true, "n_unread": 18 }</pre>
-----------------	--

api_rcv_get_first_n:

Description:	Obtains the first N received messages
---------------------	---------------------------------------

Parameters:	<table> <tr> <td>n_regs:</td> <td>Number of messages to retrieve</td> </tr> <tr> <td>set_read (optional):</td> <td>1: Mark messages retrieved as read 0: Leave the messages as they are (default)</td> </tr> </table>	n_regs:	Number of messages to retrieve	set_read (optional):	1: Mark messages retrieved as read 0: Leave the messages as they are (default)
n_regs:	Number of messages to retrieve				
set_read (optional):	1: Mark messages retrieved as read 0: Leave the messages as they are (default)				

Auth:	Yes
--------------	-----

Returns:	The result (in JSON format) of the function contains the following elements:
success:	1: Obtained all or part of the messages successfully 0: An error occurred
n_read:	If no error occurred, this field will be present and contain the number of messages successfully read from the device
reg_array:	A JSON array containing the messages. Each message has the following structure:
id:	Message ID
send_date:	Date on which the message was sent
rcv_date:	Date on which the message was received
read_date:	Date on which the message was read
numorig:	Origin number of the message
message:	Content of the message
status:	"0" means unread, "1" means message read
channel:	Device channel that received the message (*)
imsi:	Imsi that received the message (*)
error_code:	Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_rcv_get_first_n&username=lyric_api&password=lyric_api&api_version=0.05&n_
regs=1

Answer:
{ "success": true, "n_read": 1, "reg_array": [{ "id": 25, "send_date": "2010-10-20
13:05:01", "rcv_date": "2010-10-20 13:05:50", "read_date": "-", "numorig":
"12341234", "message": "hi, how are you?", "status": 0, "channel": 6, "imsi":
"123412341234" }] }

(*) Some messages may have channel "-1" and an empty imsi if they were received by any simcard on a Lyric VoIP device with firmware version below LV0.201

api_rcv_get_first_n_unread:

Description: Obtains the first N unread received messages

Parameters:

n_regs:	Number of messages to retrieve
set_read:	1: Mark messages retrieved as read 0: Leave the messages as they are

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success:	1: Obtained all or part of the messages successfully 0: An error occurred
n_read:	If no error occurred, this field will be present and contain the number of messages successfully read from the device
reg_array:	A JSON array containing the messages. Each message has the following structure: id: Message ID send_date: Date on which the message was sent rcv_date: Date on which the message was received read_date: Date on which the message was read numorig: Origin number of the message message: Content of the message status: "0" means unread, "1" means message read channel: Device channel that received the message (*) imsi: Imsi that received the message (*)
error_code:	Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_rcv_get_first_n&username=lyric_api&password=lyric_api&api_version=0.05&n_
regs=1

Answer:
{ "success": true, "n_read": 1, "reg_array": [{ "id": 25, "send_date": "2010-10-20
13:05:01", "rcv_date": "2010-10-20 13:05:50", "read_date": "-", "numorig":
"12341234", "message": "hi, how are you?", "status": 0, "channel": 6, "imsi":
"123412341234" }] }

(*) Some messages may have channel "-1" and an empty imsi if they were received by any simcard on a Lyric VoIP device with firmware version below LV0.201

api_rcv_get_last_n:

Description: Obtains the last N received messages

Parameters:

n_regs:	Number of messages to retrieve
set_read (optional):	1: Mark messages retrieved as read 0: Leave the messages as they are (default)

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success:	1: Obtained all or part of the messages successfully 0: An error occurred
n_read:	If no error occurred, this field will be present and contain the number of messages successfully read from the device
reg_array:	A JSON array containing the messages. Each message has the following structure: id: Message ID send_date: Date on which the message was sent rcv_date: Date on which the message was received read_date: Date on which the message was read numorig: Origin number of the message message: Content of the message status: "0" means unread, "1" means message read channel: Device channel that received the message (*) imsi: Imsi that received the message (*)
error_code:	Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_recv_get_last_n&username=lyric_api&password=lyric_api&api_version=0.05&n_r
egs=1

Answer:
{ "success": true, "n_read": 1, "reg_array": [{ "id": 25, "send_date": "2010-10-20
13:05:01", "recv_date": "2010-10-20 13:05:50", "read_date": "-", "numorig":
"12341234", "message": "hi, how are you?", "status": 0, "channel": 6, "imsi":
"123412341234" }] }

(*) Some messages may have channel "-1" and an empty imsi if they were received by any simcard on a Lyric VoIP device with firmware version below LV0.201

api_recv_get_last_n_unread:

Description: Obtains the last N unread received messages

Parameters:

n_regs:	Number of messages to retrieve
set_read:	1: Mark messages retrieved as read 0: Leave the messages as they are

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success:	1: Obtained all or part of the messages successfully 0: An error occurred
n_read:	If no error occurred, this field will be present and contain the number of messages successfully read from the device
reg_array:	A JSON array containing the messages. Each message has the following structure: id: Message ID send_date: Date on which the message was sent recv_date: Date on which the message was received read_date: Date on which the message was read numorig: Origin number of the message message: Content of the message status: "0" means unread, "1" means message read channel: Device channel that received the message (*) imsi: Imsi that received the message (*)
error_code:	Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_recv_get_last_n&username=lyric_api&password=lyric_api&api_version=0.05&n_recs=1

Answer:
{ "success": true, "n_read": 1, "reg_array": [{ "id": 25, "send_date": "2010-10-20 13:05:01", "recv_date": "2010-10-20 13:05:50", "read_date": "-", "numorig": "12341234", "message": "hi, how are you?", "status": 0, "channel": 6, "imsi": "123412341234" }] }

(*) Some messages may have channel "-1" and an empty imsi if they were received by any simcard on a Lyric VoIP device with firmware version below LV0.201

api_recv_set_read:

Description: Sets a message as read and its read date to the moment it is set

Parameters: id: Id of the message to be set read

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success: 1: Message set read successfully
0: An error occurred

error_code: Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_recv_set_read&username=lyric_api&password=lyric_api&api_version=0.05&id=7

Answer:
{ "success": true }

api_sms_send_enable:

Description: Enables the specified channels for SMS delivery

Parameters: chan_list: Comma separated list of channels to be enabled

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success: 1: Channels were enabled successfully
0: An error occurred

error_code: Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_sms_send_enable&username=lyric_api&password=lyric_api&api_version=0.05&c
han_list=1,2,3

Answer:
{ "success": true }

api_sms_send_disable:

Description: Disables the specified channels for SMS delivery

Parameters: chan_list: Comma separated list of channels to be disabled

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success: 1: Channels were disabled successfully
0: An error occurred

error_code: Present when success field is 0. See error codes section

Example: Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_sms_send_disable&username=lyric_api&password=lyric_api&api_version=0.05&c
han_list=1,2,3

Answer:
{ "success": true }

api_sms_delete_by_status:

Description: Deletes the messages matching the specified state from the specified queue

Parameters: dir: in: queue of the received messages
out: queue of the outgoing messages

channel (out): Specifies the target channel queue (1 to 6). If not specified, it'll use the general queue for outgoing messages

status: unread (in): deletes unread messages
read (in): deletes read messages
new (out): deletes pending messages
sent (out): deletes sent messages
failed (out): deletes messages that failed to be sent
any (in or out): deletes messages matching any status

Auth:	Yes
--------------	-----

Returns:	The result (in JSON format) of the function contains the following elements: success: 1: Message were deleted successfully 0: An error occurred error_code: Only if an error occurred. See error codes section
-----------------	---

Example:	Query: http://web_user:web_pass@192.168.200.10/cgi-bin/exec? cmd=api_sms_delete_by_status&username=lyric_api&password=lyric_api&api_version=0.0 5&dir=out&channel=5&status=sent Answer: { "success": true }
-----------------	--

api_sms_delete_by_id:

Description:	Deletes the message matching the specified id from the specified queue
---------------------	--

Parameters:	dir: in: queue of the received messages out: queue of the outgoing messages channel (out): Specifies the target channel queue (1 to 6). If not specified, it'll use the general queue for outgoing messages id: Id of the message at the specified queue
--------------------	---

Auth:	Yes
--------------	-----

Returns:	The result (in JSON format) of the function contains the following elements: success: 1: Message were deleted successfully 0: An error occurred error_code: Only if an error occurred. See error codes section
-----------------	---

Example:	Query: http://web_user:web_pass@192.168.200.10/cgi-bin/exec? cmd=api_sms_delete_by_status&username=lyric_api&password=lyric_api&api_version=0.0 5&dir=in&id=101 Answer: { "success": true }
-----------------	--

api_sms_available_space:

Description: Returns the remaining available space on the specified queue

Parameters:

dir:	in: queue of the received messages out: queue of the outgoing messages
channel (out):	Specifies the target channel queue (1 to 6). If not specified, it'll use the general queue for outgoing messages

Auth: Yes

Returns: The result (in JSON format) of the function contains the following elements:

success:	1: Message were deleted successfully 0: An error occurred
error_code:	Only if an error occurred. See error codes section
n_available:	Number of messages that can be inserted in the queue before overwriting the oldest message

Example:

Query:
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_sms_available_space&username=lyric_api&password=lyric_api&api_version=0.05
&dir=out

Answer:
{ "success": true, "n_available": 400 }

Error Codes

CODE	DESCRIPTION
DatabaseProblem	Device database couldn't be accessed
APIDisabled	Web API is disabled
DestinationEmpty	SMS destination was not specified
ContentEmpty	Content was not specified
UsernameEmpty	Username was not specified
PasswordEmpty	Password was not specified
APIVersionEmpty	API version was not specified
BadDestination	Destination must be a numeric value
BadUserPassword	Incorrect username or password
BadApiVersion	Incorrect API version
UnixSocketProblem	Connection to unix socket failed

AppAnswerProblem	Application does not answer
AppNotificationProblem	Application answer has errors
MessageIdEmpty	Message ticket was not specified
DatabaseProblemORIdNotFound	Database couldn't be accessed or ticket does not exist
DeviceNotReady	Device hasn't initiated completely yet
WrongChannel	Channel specified is invalid
MissingChannelList	A comma separated list must be specified
SetReadEmpty	Didn't specify whether to set messages as read or not
WrongSMSDir	Just "in" or "out" is accepted
NoSMSDir	sms_dir parameter was not specified
ChannelNotRequired	Specified channel for the received SMS queue
MissingStatus	Status parameter was not specified
StatusWrog	Status parameter is different from "in" or "out"
LogError	Internal error while trying to access a queue
ChannelNotInitialized	Specified channel is not initialized
ImsiNotFound	Specified imsi couldn't be found

References

-
- [1] <http://www.json.org/>
 - [2] <http://www.python.org/ftp/python/2.7/python-2.7.msi> (for Windows)
 - [3] <http://www.yx.cl/soporte.html>