Web API Documentation
Version 0.08

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 syntaxis for the queries is described below. All answers from server are in JSON format [1], which contains the result of the query sent. Arguments in the query should use URL encoding (RFC 3986 [2]). “+”s are also accepted as a form of encoding spaces.

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 aplication written in Python [3] 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 [4].

This document describes version 0.08 of the Web API.

Changelog

v0.08

- Function *api_get_status* now also returns:
  - *report_stage*: stage of the status report for the message
  - *delivery_status*: status code of the status report for the message
  - *delivery_date*: date of the last status report received for the message

v0.07

- Function *api_get_status* now also returns:
  - *last_error*: last error that occurred
  - *n_tries*: number of failed attempts to send the message
  - *num*: destination number
  - *channel*: last channel that attempted (successfully or not) to send the message
  - *send_date*: date in seconds from epoch when the message was sent
  - *recv_date*: date in seconds from epoch when the message was queued
- Added function *api_get_queue_status*

v0.06

- Added function *api_reset_queue*
- Fixed an issue with the function *api_sms_available_space* that wouldn't return the real available space
- *Fixed documentation of functions that use the “sms_dir” parameter (previously it showed “dir” instead of “sms_dir”)*

v0.05

- Fixed function *api_reset_sent_counter* not setting counters to 0 permanently
- Added the following functions:
  - *api_recv_number_unread*
  - *api_recv_get_first_n_unread*
  - *api_recv_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_recv_get_first_n* and *api_recv_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*

Parameter Enconding and Supported Characters

As it was stated in the introduction, the encoding used for the parameters is URL (RFC 3986) with the addition of “+”s that may be used for spaces.

The characters currently supported are the following:

@£$¥èùìòçØøÅå_Ææß !#¤%&'()"*+,-./0123456789:;<=>?
¡ABCDEFGHIJKLEMNOPQRSTUVWXYZÄÖÑÜ§¿abcdefghijklemnopqrstuvwxyzäöñüà

Characters like áéíóú and [] are excluded on purpose, because the use of these would truncate the maximum SMS length to 70 characters (from 160).

Web API Commands

Most of the following commands require authentication with the web server. The parameters used for authentification 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 succesfully inserted in the device database 0: An error ocurred
	message_id:	Ticket of the queued message. If an error ocurred, this field won't be present on the response
	error_code:	Only if an error ocurred. 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&destination=55555555&api_version=0.08&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 succesfully
0: An error ocurred

message_status: If an error ocurred, this field won't be present on the response.
- 0: New
- 1: Processing
- 2: Sent
- 3: Failure

last_error: If an error ocurred, this field won't be present on the response.
- 0: None
- 1: Unknown
- 2: Destination number
- 3: Content
- 4: Network, temporal
- 5: Simcard

n_tries: Number of failed attempts to send the message

num: Destination number of the message

channel: Last channel that attempted to send the message, successfully or not

send_date: Date in seconds from epoch when the message was sent

recv_date: Date in seconds from epoch when the message was queued

report_stage: Possible values are:
- 0: No status report has been received yet
- 1: Temporary status report. More reports should be expected to arrive.
- 2: Final status report. Final report received for this message.

delivery_status: Status code contained in the report. Possible values are:
- 0: Message delivered
- 1: Message forwarded by the SC but unable to confirm delivery
- 2: Message replaced by the SC
- 32: Congestion
- 35: Service rejected
- 48: Specific to each SC
- 65: Incompatible destination

delivery_date: Date of the last status report received for the message

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=0.08&channel=1

Answer:
{ "success": true, "message_status": 2, "num": "84072666", "channel": 1, "last_error": 0, "n_tries": 0, "send_date": "1336521916", "recv_date": "1336521858", "report_stage":"2", "delivery_status":"0", "delivery_date":"1336522463" }

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 ocurred

error_code: Only if an error ocurred. 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=0.08

Answer:
{ "success": true, "channels": [{ "id": 1, "sms_send_ena": 1, "state": "registered", "imsi": "73001155555555", "n_sent_sms": 5 }, { "id": 2, "sms_send_ena": 0, "state": "registered", "imsi": "73001166666666", "n_sent_sms": 0 }] }

api_get_queue_status:

Description: Obtains the status of the specified queue. It's not recommended to use this function frequently, since it checks the whole queue and may lower performance on API web intensive applications.

Parameters:

channel (optional):	Get status for queue associated with this channel. If not specified, the status returned will be from the general queue

Auth: Yes

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

success:	1: Message status was retrieved succesfully 0: An error ocurred
n_msgs:	Number of messagges on the queue
n_new:	Number of messages marked as “new”
n_proc:	Number of messages currently being processed
n_sent:	Number of messages that were sent
n_fail:	Number of messages marked as “failed”
error_code:	See error codes section

Example: Query:

```
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?
cmd=api_get_queue_status&username=lyric_api&password=lyric_api&api_version=0.08&c
hannel=1
```

Answer:

```
{ "success": true, "n_msgs": 4, "n_new": 0, "n_proc": 0, "n_sent": 4, "n_fail": 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 ocurred
error_code:	Only if an error ocurred. 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=0.08
&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": "0.06" }

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 succesfully 0: An error ocurred n_sms: If no error ocurred, 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=0.08 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 succesfully
0: An error ocurred

n_unread: If no error ocurred, 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: Query:

```
http://web_user:web_pass@192.168.200.10/cgi-bin/exec?cmd=api_recv_get_number_unread&username=lyric_api&password=lyric_api&api_version=0.08
```

Answer:

```
{ "success": true, "n_unread": 18 }
```

api_recv_get_first_n:

Description: Obtains the first 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 succesfully
0: An error ocurred

n_read: If no error ocurred, this field will be present and contain the number of messages succesfully 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_first_n&username=lyric_api&password=lyric_api&api_version=0.08&n_regs=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_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 succesfully 0: An error ocurred
n_read:	If no error ocurred, this field will be present and contain the number of messages succesfully 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_first_n&username=lyric_api&password=lyric_api&api_version=0.08&n_regs=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:

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 succesfully 0: An error ocurred
n_read:	If no error ocurred, this field will be present and contain the number of messages succesfully 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=0.08&n_regs=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 succesfully 0: An error ocurred
n_read:	If no error ocurred, this field will be present and contain the number of messages succesfully 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=0.08&n_regs=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 succesfully 0: An error ocurred
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=0.08&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 ocurred

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=0.08&chan_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 ocurred

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=0.08&chan_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:	sms_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 ocurred	
	error_code:	Only if an error ocurred. 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=0.08&sms_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:	sms_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 ocurred
	error_code:	Only if an error ocurred. 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=0.08&sms_dir=in&id=101 Answer: { "success": true }	

api_sms_available_space:

Description:	Returns the remaining available space on the specified queue	
Parameters:	sms_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 ocurred
	error_code:	Only if an error ocurred. See error codes section
	n_available:	Number of messages that can be inserted in the queue before overwritting 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=0.08&sms_dir=out Answer: { "success": true, “n_available”: 400 }	

api_reset_queue:

Description: Deletes all messages from queue if there's no message in the “Processing” state

Parameters:

sms_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: Queue was resetted succesfully

0: An error ocurred

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

Example: Query:

http://web_user:web_pass@192.168.200.10/cgi-bin/exec?cmd=api_reset_queue&username=lyric_api&password=lyric_api&api_version=0.08&sms_dir=out&channel=5

Answer:

{ "success": true }

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 completly yet
WrongChannel	Channel specified is invalid
MissingChannelList	A comma separated list must be specified

SetReadEmpty	Didn't specify whether to set messges 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
QueueProblem	Queue nonexistent or uknown queue related error

References

[1] http://www.json.org
[2] http://tools.ietf.org/html/rfc3986
[3] http://www.python.org/ftp/python/2.7/python-2.7.msi (for Windows)
[4] http://www.yx.cl/soporte.html