



# SMSHUB API WSDL

The specifications and information in this document are subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. This document may not be copied or distributed by any means, in whole or in part, for any reason, without the express written permission of RCDevs.

Copyright (c) 2010-2017 RCDevs SA. All rights reserved.

<http://www.rcdevs.com>

WebADM and OpenOTP are trademarks of RCDevs. All further trademarks are the property of their respective owners.

Limited Warranty

No guarantee is given for the correctness of the information contained in this document. Please send any comments or corrections to [info@rcdevs.com](mailto:info@rcdevs.com).

## SMSHub API Description

OpenOTP SMSOTP is only possible via Clickatell online SMS SOAP service and via RCDevs SMSHub which is a WebADM service application just like OpenOTP.

SMSHub proposes much more options than the OpenOTP basic SMS functionalities but if you use the Clickatell service only, then you do not need it.

SMSHub does not send SMS directly and will use Clickatell or another online SMS service. But with SMSHub you can:

- › Connect more online SMS services including Clickatell, AQL, OVH, Mpulse, any SMSC using the SMPP-TR protocol, and you can write your custom SMS connector scripts if you need to use other SMSCs.
- › Enable SMSC redundancy and failover by using several SMS service at the same time (with priorities).
- › Use mobile prefix -based SMS routing to select the SMSC based on mobile number prefix.
- › Send Unicode UCS-2 messages containing non-standard SMS characters such as Chinese messages.
- › Connect several OpenOTP servers to one SMSHub service.
- › Use the SMSHub API in other applications for sending user messages.

The installation of SMSHub is straightforward and only consists in running the self-installer and configure the application in WebADM.

You do not have to modify any files in the SMSHub install directory! The web services configurations are managed and stored in LDAP by WebADM. To configure SMSHub, just enter WebADM as super administrator and got to the 'Applications' menu. Click SMSHub to enter the web-based configuration.

OpenOTP SMSHub is a WebADM Web Service just like OpenOTP. It's a standalone service which will be called by OpenOTP on the same host for SMS relaying. When you setup SMSHub you should proceed this way:

1. Setup the SMS Gateway (ex. Clickatell) in OpenOTP first and get it working.
2. Install the SMSHub component correctly in your WebADM server.
3. Configure the first SMSHub SMSC connector with the same settings that worked in OpenOTP.
4. Configure SMSHub with the "STATIC" authentication backend. And set a Static Username and Static Password for the Static authentication.
5. Configure the OpenOTP SMSC with "SMSHub" and set the same static username and password you have defined in the SMSHub configuration.

Then OpenOTP will send SMS requests to the local SMSHub and will authenticate himself with the static username and password. SMSHub will relay SMS requests to the configured remote SMSC (ex. Clickatell).

You can test the SMSHub functions by using the tools directory:

```
/opt/webadm/websrvs/smsHub/bin/
```

SMSHub service logs are accessible in the Databases menu in WebADM.

Note: To be able to use LDAP authentication backend, LDAP users used for client authentication must be WebADM accounts. That means usable LDAP accounts are those containing the webadmAccount LDAP objectclass. You can enable the WebADM features on any LDAP user/group by extending it with the webadmAccount objectclass (from object extension list).

SMSHub provides a SOAP/XML and JSON API. The API specification is detailed in the smsHub.wsdl file.

SMSHub provides 3 methods:

## 1. smsHubSend

Used to send an SMS request.

The request contains the following attributes:

- > username: Service username (mandatory string).
- > password: Service password (mandatory string).
- > mobile: The destination mobile number(s) (mandatory array of string).
- > text: The message.
- > type: The SMS type (Flash or Normal).
- > sender: The sender phone number (optional string).
- > client: Client identifier (NAS) to be used in service logs (defaults to the client IP address).
- > source: IP address of the end user system (optional).
- > settings: List of SMSHub settings which will override the user/group/application server-side settings (ex. *SMSType=Flash*).

The response contains the following attributes:

- > code:
  - > 1 means SMS send success.
  - > 0 means SMS send failure.
- > error: The error ID if code 0 was returned. The ID corresponds to the error message template names in smsHub.xml (ex. AuthFailed).
- > message: The server reply message.

## 2. smssendStatus

Used to query a server status.

The request does not contain any attribute.

The response contains the following attributes:

- > status:
- > 1 if the server is willing to accept requests.
- > 2 if the server cannot accept new requests.
- > message: The server status details.

## Custom Connectors

You can define a custom SMSC connector in the sample `smsc/custom[1-4].php` files. These files are empty skeleton scripts for your custom connector(s) implementation.

The custom file contains a simple PHP function which you need to fill-in.

```
function smshub_custom1 ($msg, $mobile, $flash, $sender, $username, $password,
    $account, $url, $proxy)
```

The parameters are:

- > \$msg: The SMS message text.
- > \$mobile: The recipient mobile number. If there are several numbers per user, the function will be called several times.
- > \$flash: This is a boolean which tells if OpenOTP wants to send a flash (class0) or normal SMS.
- > \$sender: The sender number is configured in your SMSHub configuration.
- > \$username: This is the connector username from the configuration.
- > \$password: This is the connector password from the configuration.
- > account: This is the connector account from the configuration. It is generally not necessary.
- > \$url: This is the connector URL from the configuration. It is generally not necessary.
- > \$proxy: This is the WebADM HTTP proxy connection parameters. It is generally not necessary.
- > \$error: This variable allows you to return an error message which SMSHub will log to the soapd.log file.

The function must return true on success and false if there is an SMSC failure.

*This manual was prepared with great care. However, RCDevs S.A. and the author cannot assume any legal or other liability for possible errors and their consequences. No responsibility is taken for the details contained in this manual. Subject to alternation without notice. RCDevs S.A. does not enter into any responsibility in this respect. The hardware and software described in this manual is provided on the basis of a license agreement. This manual is protected by copyright law. RCDevs S.A. reserves all rights, especially for translation into foreign languages. No part of this manual may be reproduced in any way (photocopies, microfilm or other methods) or transformed into machine-readable language*

*without the prior written permission of RCDevs S.A. The latter especially applies for data processing systems. RCDevs S.A. also reserves all communication rights (lectures, radio and television). The hardware and software names mentioned in this manual are most often the registered trademarks of the respective manufacturers and as such are subject to the statutory regulations. Product and brand names are the property of RCDevs Security. © 2022 RCDevs SA, All Rights Reserved*