



Writing your first SMS application

How to create an SMS application with Aculab Cloud

Writing an SMS application with Aculab Cloud

Aculab Cloud provides a simple, easy to learn API for the rapid development of telephony applications.

Using the Aculab Cloud WebServices APIs you can rapidly build functionality within your business applications to enable you to both send and receive text messages (SMS).

This could be used to provide services including delivery updates, ETA notifications, and two-factor authentication. Such services can increase customer satisfaction as well as providing increased levels of system security.

Sending an SMS using Aculab Cloud is easy.

- You send a request containing the SMS message to Aculab Cloud using the WebServices API
- Aculab Cloud sends the SMS to the number you have specified

When sending your WebServices request to Aculab Cloud you can specify a webpage to which status updates will be sent. Aculab Cloud sends any progress updates and delivery notifications to that status page, for your application to process.

Before you start

The SMS message flow

Message flow

The diagram below shows the flow of messages between your web server, Aculab Cloud and the mobile carrier, when sending an SMS.

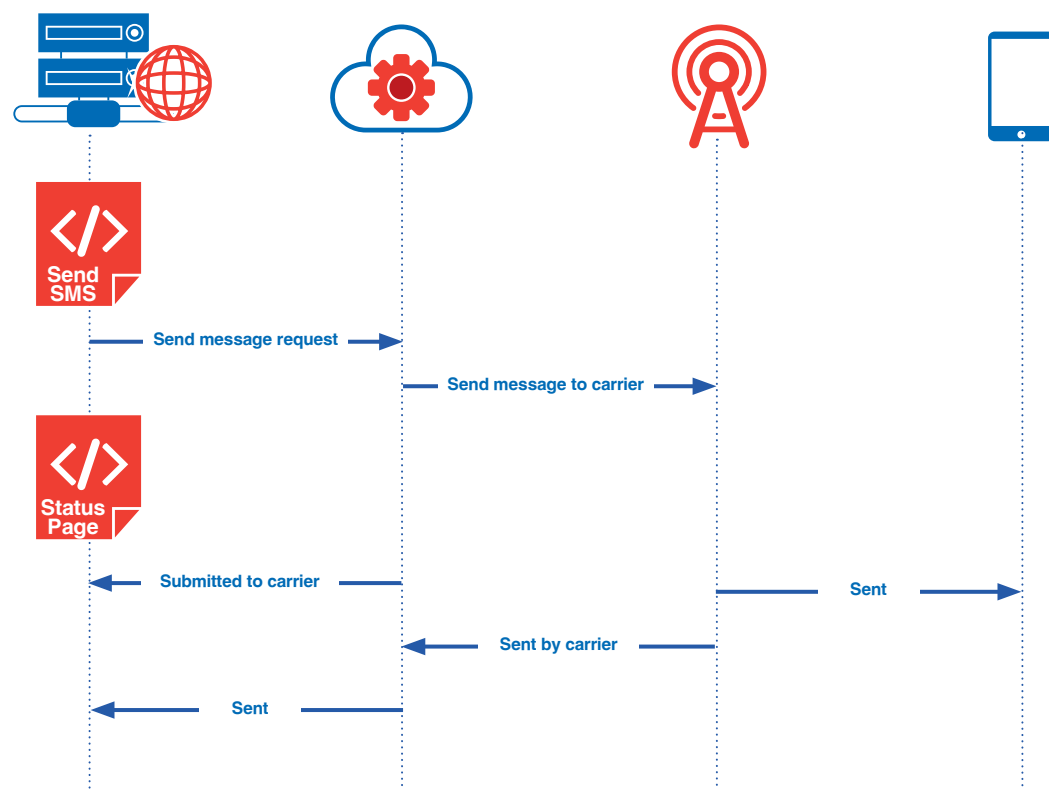


Figure 1: Outbound SMS overview

You can interact with the WebServices API directly, or you can use one of the high-level API wrappers provided by Aculab Cloud.

High-level language wrappers for Aculab Cloud's WebServices API are available in a number of languages. They can be downloaded, along with the Aculab Cloud REST API wrappers, from the Aculab Cloud website:

<https://cloud.aculab.com/>

Before you start

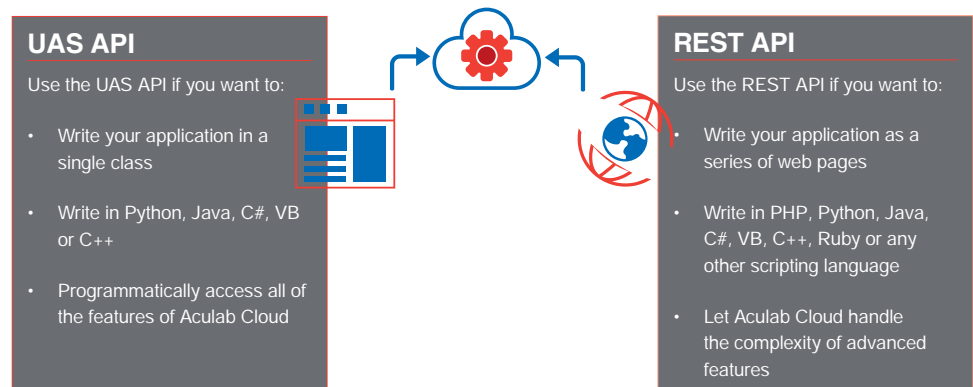
Account settings and authorisation

First Steps

1 — Decide which type of API to use

Aculab provides two methods of interacting with Aculab Cloud. When creating your account, you need to pick the UAS or REST API, but this can be changed later if you wish. The following page on the Aculab Cloud website might help with your choice:

https://cloud.aculab.com/documents/api_comparison



2 — Sign up for a free developer account

Sign up here for a free developer account on Aculab Cloud: <https://cloud.aculab.com/login/enrol>. Once you've created a developer account, a detailed quickstart guide for UAS or REST will be available.

3 — Set up test numbers

Aculab provides developers with a range of free to use test numbers for application testing. Other alternatives would be a SIP endpoint or to use WebRTC. Adding credit to your account will enable you to purchase your own inbound DID number to use with your application.

4 — Set up hosting

Decide where to host your applications and UAS code (if you choose that option). This could be one of your own servers or a web server from a cloud hosting company.

5 — Get coding!

You're now ready to begin work on Aculab Cloud.

Before you start

Account settings and authorisation

Aculab Cloud account settings

In order to send or receive SMS using your Aculab Cloud account you will need to make sure you have done the following:

- Upgraded your Aculab Cloud account to a production account (by adding funds)
- Purchased a telephone number which supports SMS
- Enabled SMS on a purchased number

You can upgrade your account and purchase an SMS enabled telephone number via the Account menu, when logged-in to your Aculab Cloud account.

WebServices API authorisation

When sending a WebServices API request to Aculab Cloud, authorisation is required. Aculab Cloud will ask for your WebServices API username and password. The username has the format cloud-id/cloud-username and the password is the password you use to log in to your Aculab Cloud account.

For example, if you were sending the request via the European cloud (0-2-0), and your username was me@my_email.com, then your WebServices API username would be [0-2-0/me@my_email.com](#)

The subsequent sections of this document shall demonstrate the construction of web pages to both send an SMS, and process SMS status updates.

For this code walkthrough the PHP wrapper for Aculab Cloud's WebServices API will be used. Wrappers for other languages are available via the Aculab Cloud website.

Sending an SMS

Creating an SMS application with Aculab Cloud

Sending an SMS

Using the Aculab Cloud high-level wrappers makes it even easier to add SMS functionality into your existing applications. An example of sending an SMS using the PHP wrapper can be seen below.

```
<?php
declare(encoding='UTF-8');
spl_autoload_register();

use \Aculab\TelephonyRestAPI\MessageWebService;
use \Aculab\TelephonyRestAPI\Message;

$MyService = new MessageWebService( CLOUD_ID, CLOUD_USERNAME, CLOUD_PASSWORD);

# Details for contents of my message
$from = "111111111111";
$to = "222222222222";
$text = "Hello world";

$MyMessage = new Message($from, $to, $text);

# Set the page to which status updates will be sent
$MyMessage->setStatusPage('http://myserver.com/sms_status.php');

$MyService->sendMessage($MyMessage);
?>
```

Figure 2: Sending an SMS using the PHP wrapper

Walking through this code, the first thing to do is set the character encoding to Unicode, which allows for international characters in any text.

```
declare(encoding='UTF-8');
```

`spl_autoload_register()` is used to auto-load any classes defined within the included Aculab classes.

```
spl_autoload_register();
```

Sending an SMS

Creating an SMS application with Aculab Cloud

Next, import the Aculab classes that are going to be used within the webpage.

```
use \Aculab\TelephonyRestAPI\MessageWebService;  
use \Aculab\TelephonyRestAPI\Message;
```

Before sending a message a MessageWebService must be created. This is an object through which any requests associated with SMS are sent to Aculab Cloud. When creating this object, the cloud ID of the Aculab Cloud region we want to send messages from is passed, along with the user credentials for our Aculab Cloud account.

```
$MyService = new MessageWebService( CLOUD_ID, CLOUD_USERNAME, CLOUD_PASSWORD);
```

Next, create the message which you wish to send, including the originating and destination phone numbers, and the text contents of the SMS message.

The originating phone number **must** be an SMS enabled number on your Aculab Cloud account.

```
# Details for contents of my message  
$from = "11111111111";  
$to = "22222222222";  
$text = "Hello world";  
  
$MyMessage = new Message($from, $to, $text);
```

In this example we have also set a page to which SMS status updates should be sent.

```
# Set the page to which status updates will be sent  
$MyMessage->setStatusPage('http://myserver.com/sms_status.php');
```

All that is left to do is to ask Aculab Cloud to send the message.

```
$MyService->sendMessage($MyMessage);
```

Sending an SMS

Creating an SMS application with Aculab Cloud

You can easily test this PHP code by placing it on your own webserver, editing the file to use your cloud credentials, and then requesting the page via your web browser.

If sending the request to Aculab Cloud was successful, then you will receive a 200 OK response along with some JSON encoded data containing a reference ID. For example:

```
["2e891aea_0.1366641192.77832"]
```

This reference ID can be used in other SMS WebServices APIs, to get the current status of the SMS message, or to cancel sending of the SMS if it has not yet been sent.

Handling SMS status updates

Creating an SMS application with Aculab Cloud

Handling SMS status updates

As mentioned earlier, it is possible to specify a status page when sending an SMS. This is a page to which progress updates on the status of your SMS will be sent.

One possible status update is reporting delivery success. Requesting of delivery reports is enabled on a per message basis.

By setting 'request_delivery_report' to true, once the SMS carrier reports the SMS has been delivered, a specified status page will be requested from your webserver, and the status will be shown as 'delivered'.

Using the delivery report feature could be very useful for confirming that numbers being used for features such as appointment reminders are valid, and that reminders have arrived at a customer's handset.

Below is an example page which could be used for processing SMS status updates.

```
<?php
declare (encoding='UTF-8');
spl_autoload_register();

use \aculab\telephonyrestapi\MessageStatusUpdate;

$update = MessageStatusUpdate::getMessageStatusUpdate();

// Retrieve contact details
$to = $update->getTo();
$from = $update->getFrom();
$status = $update->getStatus();

error_log("From: ".$from." To: ".$to." Status: ".$status."\n");
?>
```

Figure 3: Processing SMS status updates

Stepping through the code above, the required Aculab classes are imported:

```
use \aculab\telephonyrestapi\MessageStatusUpdate;
```

The status update is then retrieved:

```
$update = MessageStatusUpdate::getMessageStatusUpdate();
```


Handling SMS status updates

Creating an SMS application with Aculab Cloud

Finally the status update to your web server logging is printed out:

```
// Retrieve contact details
$to = $update->getTo();
$from = $update->getFrom();
$status = $update->getStatus();

error_log("From:". $from. " To:". $to. " Status:". $status. "\n");
```

In a real world application you may want to store the status information in a secure database, or write the SMS progress to a webpage to give a visual update of when an SMS has been sent.

That completes the walkthrough of this outbound SMS application.

What to do next?

This walkthrough has been focused on PHP. However, Aculab provides additional wrappers around its REST API in .NET, Java, Python and Ruby.

Code samples for the other wrappers can be found on the Aculab Cloud website. <https://cloud.aculab.com>

If you have any questions, you can email Aculab's support team via support@aculab.com

Aculab Cloud

Three decades of innovation - in the cloud

About Aculab

Aculab provides deployment proven telephony products to the global communications market

Whether you need telephony resources on a board, on a host server processor or from a cloud-based platform, Aculab ensures that you have the choice. We are an innovative, market leading company that places product quality and support right at the top of our agenda. With over 35 years of experience in helping to drive our customers' success, our technology is used to deliver multimodal voice, data and fax solutions for use within IP, PSTN and mobile networks – with performance levels that are second to none.

For more information

To learn more about Aculab Cloud and Aculab's extensive telephony solutions visit:

www.aculab.com

www.aculab.com

AMERICAS

+1 (781) 352 3550 | sales@aculab.com

EMEA

+44 (0) 1908 273802 | sales@aculab.com

Leverage the heritage of Aculab when you move to the cloud

Moving your application development environment to a cloud infrastructure is a big step. Despite the clear benefits of cloud migration, it's natural for developers of hardware-based solutions to be concerned about the risks of moving their technology IP – and the years of investment and knowledge that has gone into creating it – to a new cloud development platform. Most of the big names in cloud communications are relatively new entrants to the communications market; some are working with open source technologies and, as the market consolidates, it is likely that many will not be in business in just a few years' time. So how do you know that a cloud platform can deliver the same level of reliability and performance that you've come to expect from a hardware deployment, and that it will be around for decades?

Three decades of innovation — the next chapter

Aculab Cloud deploys Aculab's industry benchmark technology and has been built organically out of more than 35 years' worth of experience in the communications enablement market. Put simply, it's the result of more than three decades of experience and innovation.

Aculab Cloud developers can be assured that the technology that powers Aculab Cloud has been used to enable tens of thousands of mission-critical applications

across the world. Aculab Cloud features robust, field-proven protocols that have been developed and honed in conjunction with thousands of developers and deployed across hundreds of networks.

It's the only cloud communications platform that delivers the expertise, experience and reliability that you get from working with a proven communications enabler.

Leverage our heritage when you move to the cloud.