Acapela Cloud

Vaas to Acapela Cloud Migration

Acapela Cloud Migration procedure manual

Published 16 April 2020

This document was produced by Acapela Group. We welcome and consider all comments and suggestions. Please, use the *Contact Us* link at our website: http://www.acapela-group.com

Table of contents

Definitions

API : Application Program Interface

SSL : Secure Sockets Layer, secure protocol developed for sending information securely over the Internet.

TTS : Text-to-Speech

Vaas : Voice as a Service (www.acapela-vaas.com)

1. Preamble

1.1. Product information

Acapela Cloud is the new solution for developing online applications with Acapela voices.

It replaces the Acapela Vaas (Voice as a service) solution and has been completely redesigned to offer a modern approach.

The key benefits of this new product are:

- It is based on modern frameworks and the whole associated website/documentation has been revised with an improved interface.
- It offers an easy to use and integrate API
- It uses the latest protocols & standards at the highest security levels like SSL support.
- It provides new features like a better lexicon handling, additional formats for audio, support for the latest version of voices ...
- It has been optimised to improve response time and reduce latency
- It allows improved control and management of audio consumption with statistics and reporting functionalities
- More than 100 voices in over 30 languages are available 24/7

1.2. About this documentation

The purpose of this document is to present the different steps of the migration of an application from Vaas to Acapela Cloud.

It is not intended to describe the functions of the API. For this, the reference documentation is available online directly accessible from the Acapela Cloud website https://www.acapela-cloud.com/docs/.

2. Migration from Vaas

2.1. How to start with Acapela Cloud

Once you have an Acapela Cloud account with audio credits, you can start and in 2 steps generate audio.

2.1.1. Login

The first step is to connect to the service with your credentials by calling the ***/api/login*** function and retrieve the token that will be used in all subsequent calls.

This token is used to identify you and obtain your authorizations.

2.1.2. Speak

Once you have received your token, the next step is to call the main function of the API: ***/api/command***. This request generates TTS and returns the audio as a stream or as an audio file.

2.2. From Vaas to Acapela Cloud

2.2.1. Identification

When using Vaas, the identification information must be included in each request (see example below).

When using Acapela Cloud, once you are logged in, you only have to include the returned token in your request.

2.2.2. Speak request

A speak request requires a few mandatory parameters: identification, voice, text, type of audio generation and format of the audio output.

- The identification:

Vaas: **cl_login / cl_pwd and cl_app**

The 3 parameters must be included in each request.

Acapela Cloud: **token**

This is the value generated when calling the login function.

- The name of the voice:

Vaas: **req_voice** (req_voice=sharon22k)

The format is the name of the voice written in lowercase + 22k

Acapela Cloud: **voice** (voice= Sharon22k_HQ)

The format is the name of the voice starting with a capital letter + 22k_HQ

Note: be careful to the format and the case of the voice name.

- The URL encoded text to be synthesized:

Vaas: **req_text** (req_text = ”Hello, how are you ?”)

Acapela Cloud: **text** (text = ”Hello, how are you ?”)

- The type of audio generation:

Vaas: **req_asw_type** (req_asw_type=STREAM)

The default value “INFO” is not generally used and is replaced by STREAM or SOUND

Acapela Cloud: **output** (default output=stream)

- The audio format:

Vaas: **req_snd_type** (default req_snd_type = MP3)

Acapela Cloud: **type** (default type = mp3)

2.2.3. Parameters

Here is correspondence table for the main parameters of a speak request:

Parameter	Vaas	Acapela Cloud
Identification	**cl_login / cl_pwd / cl_app**	**token**
Voice name	**req_voice** ***for example: sharon22k***	**voice** ***for example: sharon22k_HQ***
Text	**req_text** ***URL encoded text***	**text** ***URL encoded text***
Type of audio output	req_asw_type *INFO(default)/SOUND/STREAM*	output *stream (default)/storage*
Audio format	req_snd_type *MP3(default)/OGG/WAV/RAW*	type *mp3 (default)/ogg/wav*

Word position	req_wp “” (default) / “on”	wordpos “off” (default) / “on”
Mouth position	req_mp “” (default) / “on”	mouthpos “off” (default) / “on”
Speed	req_spd *60 to 360 (default 180)*	speed 30 to 300 (default 100)
Voice shaping	req_vct *50 to 150 (default 100)*	shape 50 to 150 (default 100)
Volume	req_vol *50 to 65535 (default 32768)*	volume 50 to 65535 (default 32768)
User lexicon	req_lexicon (obsolete)	dico *Lexicon previously uploaded on Acapela Cloud customer account (for example: Dico.dic)*
Application identifier	cl_app (part of the identify parameter)	application Application name

For more parameters and options, you can go to the reference documentation online.

Note:

Some presets have been defined in Vaas like:

- /streamer.mp3 for Synthesizer?req_asw_type=STREAM&req_snd_type=mp3
- /filemaker.ogg for Synthesizer?req_asw_type=SOUND&req_snd_type=ogg

There is no equivalence in Acapela Cloud.

3. Samples

3.1. Vaas: javascript speak sample

```
$(document).ready(function()
{
// Request(stream)
      var url = "http://vaas.acapela-group.com/Services/Synthesizer?" +
      "cl_login=xxx&cl_app=yyy&cl_pwd=zzz" +
      "&req_asw_type=STREAM&req_snd_type=MP3" +
      "&req_text=Hello, how are you?&req_voice=rachel22k";

// Play audio
      var output = '<audio controls="controls" ' +
      'autobuffer="autobuffer" autoplay="autoplay">' +
      '<source src="' + url + '" type="audio/mp3" /></audio>';

      $('#audioSynth').html(output);
});
```

3.2. Acapela Cloud: javascript speak sample

```
$(document).ready(function()
{
//login
      $.ajax({
            url: "https://www.acapela-cloud.com/api/login/",
            type: "POST",
            dataType: "json",
            data: {
                  email: "xxx",
                  password: "yyy"
            },
            success: function (data) {
                  token = data.token;

// Request (stream)
                  var url = "https://www.acapela-cloud.com/api/command/" +
                  "?voice=RachelTransport22k_HQ&text=Hello, how are you?" +
                  "&output=stream&type=mp3&token=" + token;
// Play audio
                    var output = '<audio controls="controls" ' +
                    'autobuffer="autobuffer" autoplay="autoplay">' +
                    '<source src="' + url + '" type="audio/mp3" /></audio>';

                    $('#response').html(output);
                }
            }).fail(function (data) {
                alert(data.responseText);
            });
});
```

3.3. More Acapela Cloud examples

Examples in different programming languages can be downloaded on the Acapela Cloud website from the documentation page: https://www.acapela-cloud.com/docs/