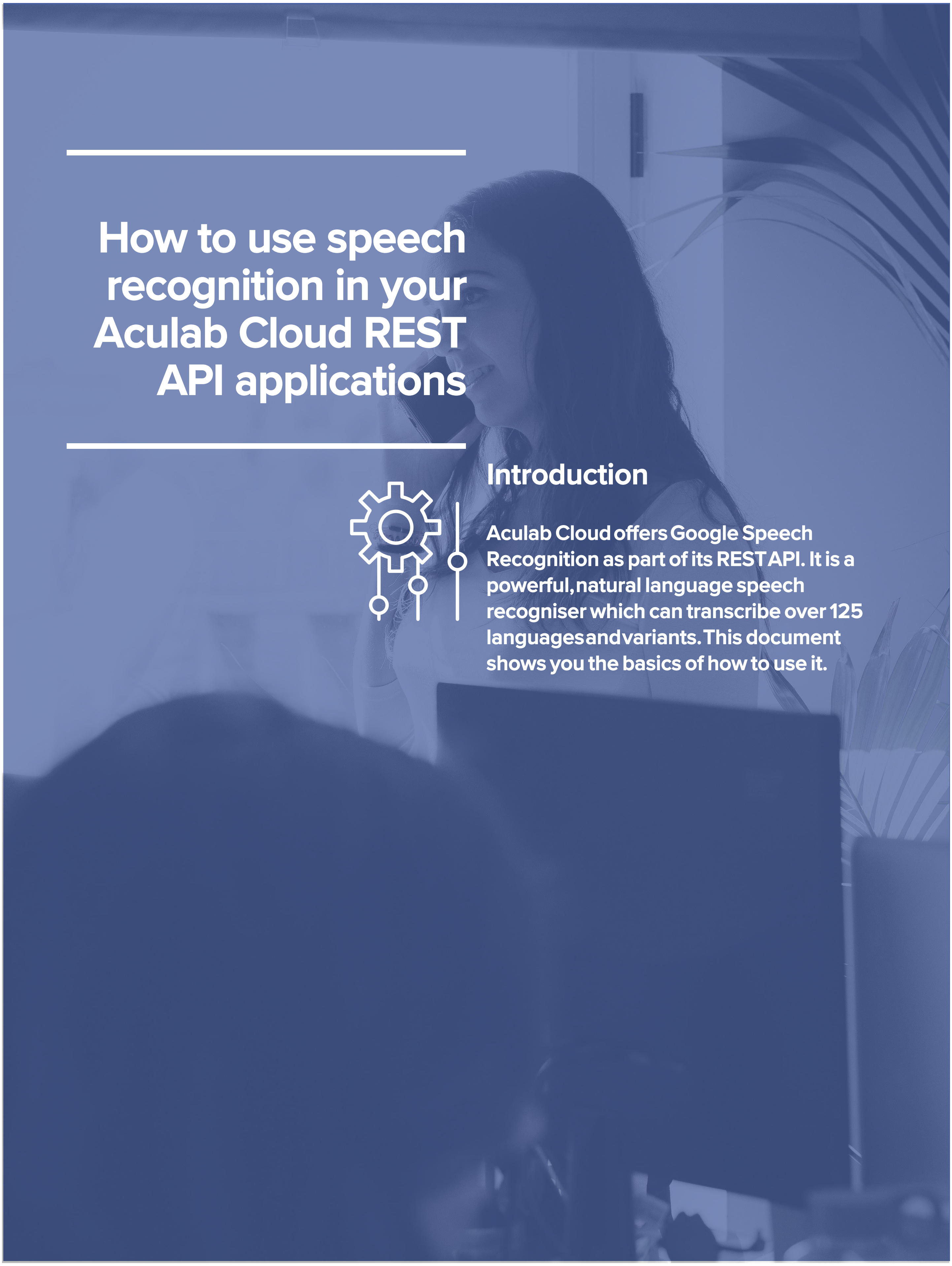

How to use speech recognition in your Aculab Cloud REST API applications

Introduction

Aculab Cloud offers Google Speech Recognition as part of its REST API. It is a powerful, natural language speech recogniser which can transcribe over 125 languages and variants. This document shows you the basics of how to use it.

How to use speech recognition in your Aculab Cloud REST API applications

Ways you can use speech recognition

There are two ways in which you can use speech recognition during calls.

- As part of the call dialog:
    - Asking a caller for user input, and navigating a voice menu system.
- To transcribe a call live, independent of the call dialog:
    - Providing a live text transcription of a call to an agent, or to a system that advises the agent based on what's been said.

Part of the call dialog

Let's dive in to using speech recognition as part of the call dialog, and consider how we'd code an Aculab Cloud REST API v2 application to do it. We'll use php in this document, but we also have wrappers in a range of other languages which you can download from your Aculab Cloud account.

We'll focus on the get_input action – see:

https://www.aculab.com/cloud/voice-and-fax-apis/rest-api/rest-api-version-2/actions/get-input.

Starting the recognition

The simplest code of all is below:

```
$prompt = Aculab\TelephonyRestAPI\Play::sayText('What would you like to do?');
$getInput = new Aculab\TelephonyRestAPI\GetInput('GetInputPage.php');
$getInput->setPrompt($prompt);
$response->addAction($getInput);
```

Here, we create a Play object, which uses Text-to-Speech (TTS) to ask the caller a question. Then, we create a GetInput object, passing it a webhook, next_page, which is where the result of the get_input action is sent when completed successfully. Then, we pass the Play object we just created as the prompt and, finally, we add the GetInput object to the list of actions we'll return in our response. We could have specified the language to be recognised, but we are going to use the default en-US for now.

Getting the result

The recogniser will listen to audio from the caller and, when they've finished talking, POST to the webhook you provided, GetInputPage.php in our example, passing it the results as a JSON object. The speech results provided are pretty detailed but, for now, we'll simply use them to create a string containing the recogniser's final view on what was said:

```
$phrases = $getInputResult->getSpeechInput();
$phrase = array_shift($phrases);
if ($phrase->getFinal()) {
    $dir = $phrase->getDirection();
    $alternatives = $phrase->getAlternatives();
    $speech = array_shift($alternatives);
    if ($speech) {
        $text = $speech->getText();
    }
}
```

How to use speech recognition in your Aculab Cloud REST API applications

The webhook is passed an array of phrase objects, each consisting of alternatives (an array of one or more speech objects), itself each containing the text that was recognised and an optional confidence level. The speech objects are ordered most confident first, so the code above can simply run through the phrases, appending the text from the first of any alternatives. Then, you'd add code to interpret the text and move the dialog on to the next step.

On another note

It is also possible to allow the caller to barge-in on a long running play only when certain words are said.

This can be useful, for example, when playing a voicemail and only allowing it to be interrupted by a key word like 'cancel' or 'stop'.

For information, see the play action, https:/www.aculab.com/cloud/voice-and-fax-apis/rest-api/rest-api-version-2/actions/play, and its barge_in_on_speech property.

Transcribe a call live

Next, we'll see how to code an Aculab Cloud REST API v2 application to transcribe a call live, independent of the call dialog. We'll be using the start_transcription action, https://www.aculab.com/cloud/voice-and-fax-apis/rest-api/rest-api-version-2/actions/start-transcription.

Starting the recognition

The simplest code of all is below:

```
$start = new Aculab\TelephonyRestAPI\StartTranscription('TranscriptionResults.php');
$start->setTranscriptionMode('inbound');
$response->addAction($start);
```

We create a StartTranscription object, tell it to listen to the inbound audio from the caller, and pass it the webhook, result_page, to which results are sent as the recogniser listens to speech. This will be an ongoing recognition which continues until we stop it (or the call ends) and which runs independent of the dialog. So, we can run other actions while it's recognising. We then simply add the object to the list of actions we'll return in our response. Again, we'll leave the language at the default, en-US.

Getting the result

The recogniser will listen to audio as configured, and POST to the results_page provided, TranscriptionResults.php in our example, passing it the results. The speech results are presented in much the same way as before but, this time, we expect our page to be called on an ongoing basis, each time with more text from the recogniser as new phrases are spoken:

```
$text = '';
foreach ($transcriptionResult->getTranscription() as $phrase) {
   if ($phrase->getFinal()) {
      $alternatives = $phrase->getAlternatives();
      $speech = array_shift($alternatives);
      if ($speech) {
         $text = $text . $speech->getText();
      }
   }
}
```

How to use speech recognition in your Aculab Cloud REST API applications

Hinting the recognition

It is also possible to provide an array of hints to the recogniser. This allows you to improve its performance by providing hints of what's likely to be spoken. For get_input, this would be provided via the speech_recognition_options word_hints property.

A word hint is an array of strings, each containing a word, a phrase or a class token, and also a priority (reserved for future use). As well as the obvious hints, for example 'red', 'green', and 'blue' if you're recognising the primary colours, you can also provide class tokens. These allow you to specify that more complex things are likely to be said, for example $FULL_PHONENUM to represent a phone number and $MONEY to represent an amount with currency.

See https://www.aculab.com/cloud/voice-and-fax-apis/rest-api/rest-api-version-2/support-classes/word-hint for more details, including a link to Google's site where you can find a list of these tokens.

Further reading

A general description of the speech recognition functionality available to use with Aculab Cloud is provided here:

https://www.aculab.com/cloud/guides/speech-recognition.

Appendix

If you'd like to get basic working applications covering the use cases above, download the PHP wrappers and see the simple_get_input and simple_start_transcription samples.

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

Contact us

Phone
+44 (0) 1908 273800 (UK)
+1 (781) 352 3550 (USA)

Email
info@aculab.com
sales@aculab.com
support@aculab.com

Social
@aculab
aculab