Google Cloud Speech API: Qwik Start

Overview

The Google Cloud Speech API enables easy integration of Google speech recognition technologies into developer applications. The Speech API allows you to send audio and receive a text transcription from the service (see What is the Google Cloud Speech API? for more information).

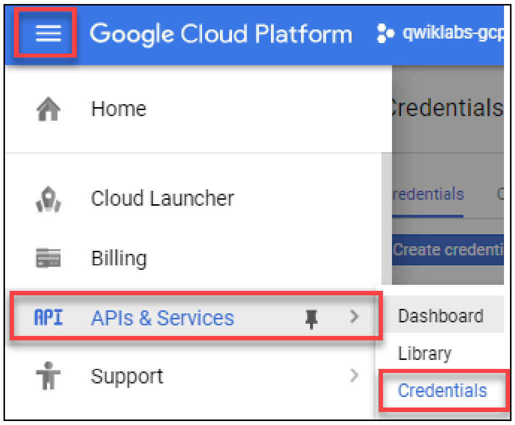
What you'll do

- Create an API key
- Create a Speech API request
- Call the Speech API request

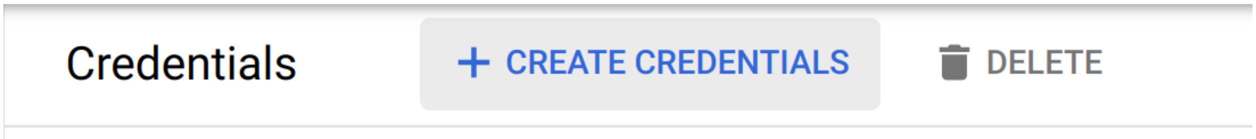
Create an API Key

Since you'll be using `curl` to send a request to the Speech API, you'll need to generate an API key to pass in our request URL.

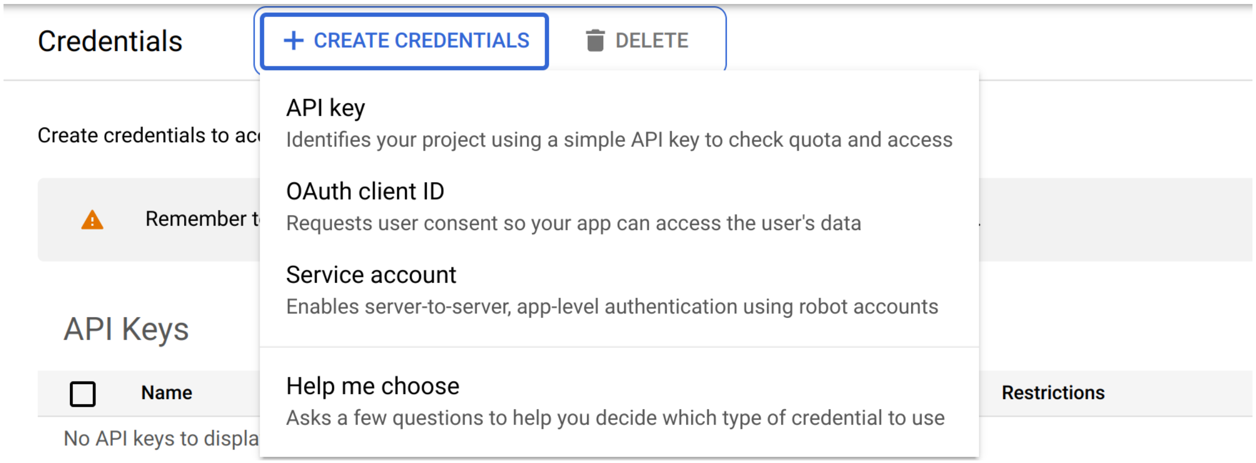
To create an API key, click **Navigation menu** > **APIs & services** > **Credentials**:



Then click **Create credentials**:



In the drop down menu, select **API key**:



API key
Identifies your project using a simple API key to check quota and access

OAuth client ID
Requests user consent so your app can access the user's data

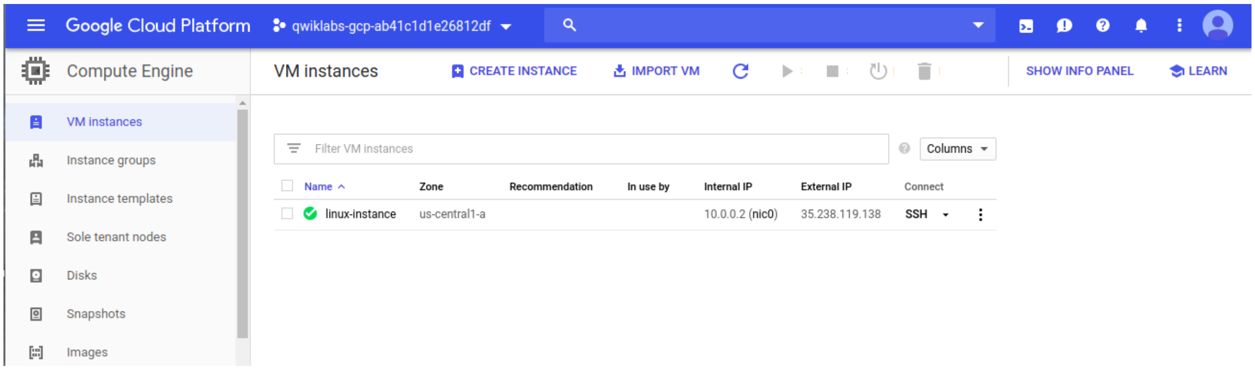
Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

Copy the key you just generated and click **Close**.

Now that you have an API key, you will save it as an environment variable to avoid having to insert the value of your API key in each request.

In order to perform next steps please connect to the instance provisioned for you via ssh. Open the **Navigation menu** and select **Compute Engine**. You should see the following provisioned linux instance:



Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
linux-instance	us-central1-a			10.0.0.2 (nic0)	35.238.119.138	SSH

Click on the **SSH** button. You will be brought to an interactive shell. In the command line, enter in the following, replacing `<YOUR_API_KEY>` with the key you just copied:

```
export API_KEY=<YOUR_API_KEY>
```

Remain in this SSH session for the rest of the lab.

Create your Speech API request

Note: You will use a pre-recorded file that's available on Cloud Storage: `gs://cloud-samples-tests/speech/brooklyn.flac`. You can listen to this file before sending it to the Speech API here.

Create `request.json` in SSH command line. You'll use this to build your request to the speech API:

```
touch request.json
```

Now open the `request.json`:

```
nano request.json
```

You can use your preferred command line editor (`nano`, `vim`, `emacs`) or `gcloud`. This lab will provide instructions for nano.

Add the following to your `request.json` file, using the `uri` value of the sample raw audio file:

```
{
  "config": {
      "encoding":"FLAC",
      "languageCode": "en-US"
  },
  "audio": {
      "uri":"gs://cloud-samples-tests/speech/brooklyn.flac"
  }
}
```

Press `control` + x and then y to save and close the `request.json` file.

The request body has a `config` and `audio` object.

In `config`, you tell the Speech API how to process the request:

- The `encoding` parameter tells the API which type of audio encoding you're using while the file is being sent to the API. `FLAC` is the encoding type for .raw files (here is documentation for encoding types for more details).

There are other parameters you can add to your `config` object, but `encoding` is the only required one.

In the `audio` object, you pass the API the uri of the audio file in Cloud Storage. Now you're ready to call the Speech API!

Call the Speech API

Pass your request body, along with the API key environment variable, to the Speech API with the following `curl` command (all in one single command line):

```
curl -s -X POST -H "Content-Type: application/json" --data-binary
@request.json
"https://speech.googleapis.com/v1/speech:recognize?key=${API_KEY}"
```

Your response should look something like this:

```
{
  "results": [
    {
      "alternatives": [
        {
          "transcript": "how old is the Brooklyn Bridge",
          "confidence": 0.98267895
        }
      ]
    }
  ]
}
```

The `transcript` value will return the Speech API's text transcription of your audio file, and the `confidence` value indicates how sure the API is that it has accurately transcribed your audio.

You'll notice that you called the `syncrecognize` method in the request above. The Speech API supports both synchronous and asynchronous speech to text transcription. In this example you sent it a complete audio file, but you can also use the `syncrecognize` method to perform streaming speech to text transcription while the user is still speaking.

You created a Speech API request then called the Speech API. Run the following command to save the response in a `result.json` file:

```
curl -s -X POST -H "Content-Type: application/json" --data-binary
@request.json
"https://speech.googleapis.com/v1/speech:recognize?key=${API_KEY}" >
result.json
```

Congratulations!

This concludes the self-paced lab, Google Cloud Speech API: Qwik Start. You integrated speech recognition into an app, and then generated transcription from the service.