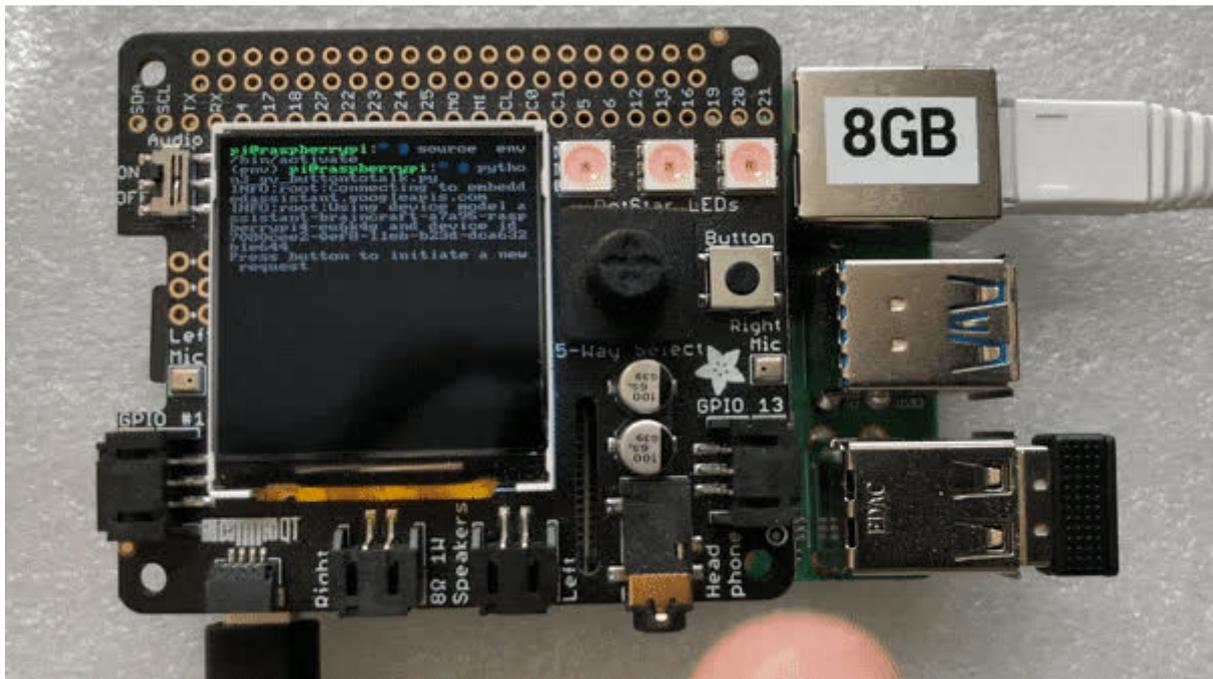




# Using Google Assistant on the BrainCraft HAT or Voice Bonnet

Created by Melissa LeBlanc-Williams



<https://learn.adafruit.com/using-google-assistant-on-the-braincraft-hat>

Last updated on 2024-06-03 03:14:30 PM EDT

# Table of Contents

<b>Overview</b>	<b>3</b>
<ul style="list-style-type: none"><li>• Parts</li></ul>	
<b>Google Setup</b>	<b>6</b>
<ul style="list-style-type: none"><li>• Project Creation</li><li>• Device Registration</li><li>• Google Assistant API Setup</li><li>• Add Test Users</li><li>• Enabling Permissions</li></ul>	
<b>Device Setup</b>	<b>13</b>
<ul style="list-style-type: none"><li>• Generating an OAuth Token</li></ul>	
<b>Usage</b>	<b>17</b>
<ul style="list-style-type: none"><li>• Running the Google Assistant Script</li><li>• Associating your Project ID and Model ID</li><li>• Running the Project Script</li><li>• Phrases to Try</li></ul>	

---

# Overview

Have you ever thought about setting up your own Google Assistant device that you can customize? With the Raspberry Pi and BrainCraft HAT, you can do just that.

In this tutorial, we will walk you through setting up the Google Assistant API. Once that's all set up, you can install a few library, enable permissions and get the Google Assistant running on the Pi. Now you can ask Google what you want with the simple push of a button.



## Parts

There are a few basic parts that are needed for this project. To start with, You can use a BrainCraft HAT which includes a display.

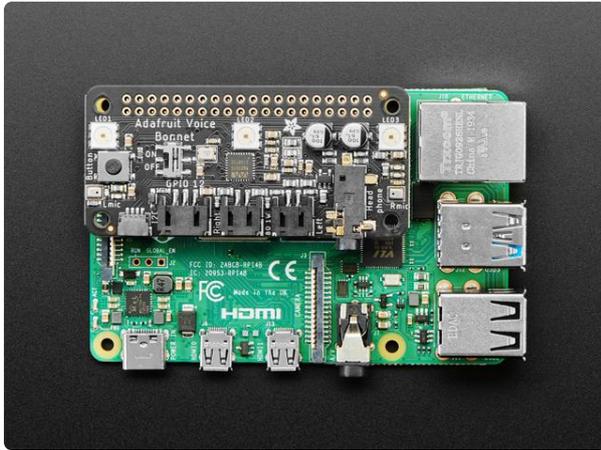


[Adafruit BrainCraft HAT - Machine Learning for Raspberry Pi 4](https://www.adafruit.com/product/4374)

The idea behind the BrainCraft HAT is that you'd be able to "craft brains" for Machine Learning on the EDGE, with Microcontrollers & Microcomputers. On ASK...

<https://www.adafruit.com/product/4374>

Or you can use the Voice Bonnet.



### Adafruit Voice Bonnet for Raspberry Pi - Two Speakers + Two Mics

Your Raspberry Pi computer is like an electronic brain - and with the Adafruit Voice Bonnet you can give it a mouth and ears as well! Featuring two microphones and two...

<https://www.adafruit.com/product/4757>

The rest of the parts are for either of the above options.



### Raspberry Pi 4 Model B - 2 GB RAM

The Raspberry Pi 4 Model B is the newest Raspberry Pi computer made, and the Pi Foundation knows you can always make a good thing

<https://www.adafruit.com/product/4292>



### Official Raspberry Pi Power Supply 5.1V 3A with USB C

The official Raspberry Pi USB-C power supply is here! And of course, we have 'em in classic Adafruit black! Superfast with just the right amount of cable length to get your Pi 4...

<https://www.adafruit.com/product/4298>



### Mono Enclosed Speaker - 3W 4 Ohm

Listen up! This 2.8" x 1.2" speaker is a great addition to any audio project where you need 4 ohm impedance and 3W or less of power. We particularly like...

<https://www.adafruit.com/product/3351>



### SD/MicroSD Memory Card (8 GB SDHC)

Add mega-storage in a jiffy using this 8 GB class 4 micro-SD card. It comes with a SD adapter so you can use it with any of our shields or adapters. Preformatted to FAT so it works out...

<https://www.adafruit.com/product/1294>



### Monoprice 5-Watt Guitar Amplifier, Portable Recorder

Always be ready rock out or record with this 5-watt Guitar Amplifier, Portable Recorder, and USB Audio Interface from Monoprice. This handheld 5-watt amp...

<https://www.adafruit.com/product/4402>



### Cell-phone TRRS Headset - Earbud Headphones w/ Microphone

These earbud headphones are the perfect accessory for your FONA - they've been tested to work with our modules - but can be used with any iOS or Android device that uses a TRRS...

<https://www.adafruit.com/product/1966>

---

# Google Setup

We will need to configure a new Google project and set up everything before we move over to the Raspberry Pi.

## Project Creation

First, you'll need to configure a new Actions Console project.

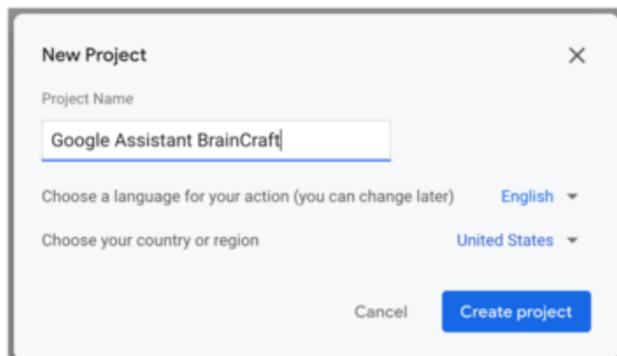
---

You don't have any projects yet. Click below to get started.



Start by going to <https://console.actions.google.com/> (<https://adafru.it/O8C>) and clicking **New Project**

---

A screenshot of the "New Project" dialog box in the Google Actions Console. The dialog has a title bar with "New Project" and a close button (X). Below the title bar, there is a "Project Name" label and a text input field containing "Google Assistant BrainCraft". Below the input field, there are two dropdown menus: "Choose a language for your action (you can change later)" set to "English" and "Choose your country or region" set to "United States". At the bottom of the dialog, there are two buttons: "Cancel" and "Create project".

Enter a Project name such as **Google Assistant BrainCraft** and click on **Create Project**

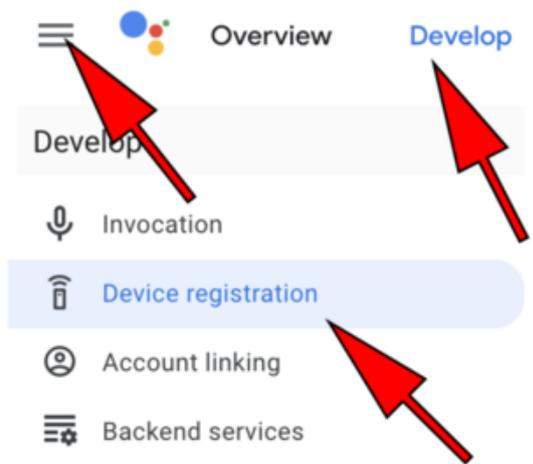


Are you looking for device registration? [Click here](#)

At the bottom of the page, click on the link to go to Device Registration.

---

## Device Registration



If you have accidentally closed the window, you can get to Device Registration by going back to the Actions Console and Selecting your project. After that, click on the **Develop** tab, the **3-Bar Menu** in the upper left, and the **Device registration** menu item.



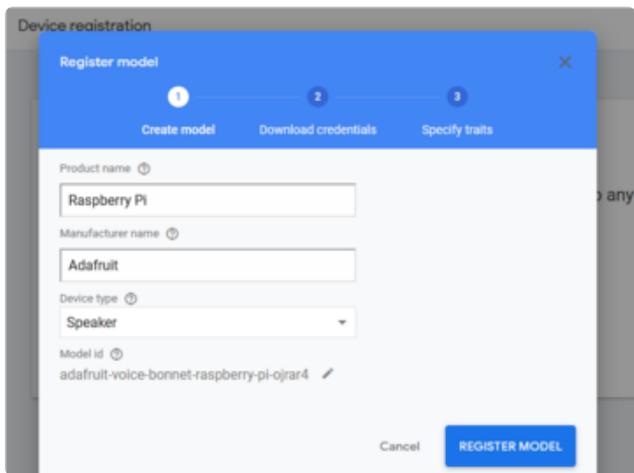
Embed the Google Assistant into any of your devices.

[Learn more](#)

**REGISTER MODEL**

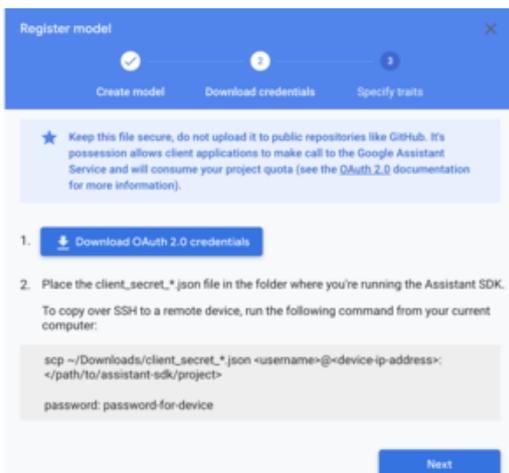
At the device registration, click the **Register Model** button.

Take note of your Model ID in this next step. You will need this in a later step.

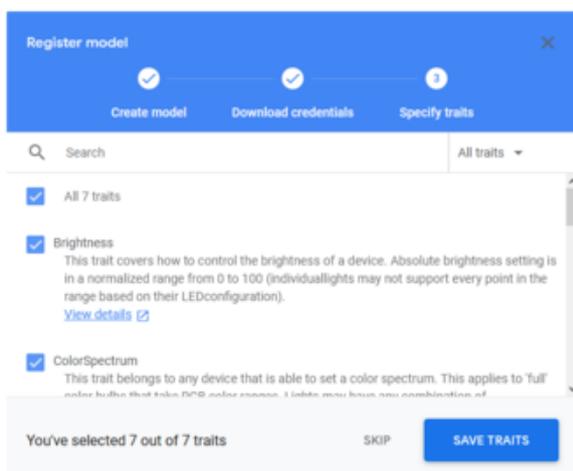


Fill in the fields with the requested information. For device type, choose **Speaker**. Make note of your **Model ID** and save it in a safe place. Click the **Register Model Button**.

If you forgot to copy your **Model ID**, don't worry. This guide will show you how to retrieve it when you need it.



Click **Download OAuth 2.0 Credentials** and save the JSON file and put it in a safe place such as your Desktop. Save the file as **client-secret.json**. You will upload this to your Raspberry Pi in a later step. Click **Next**.

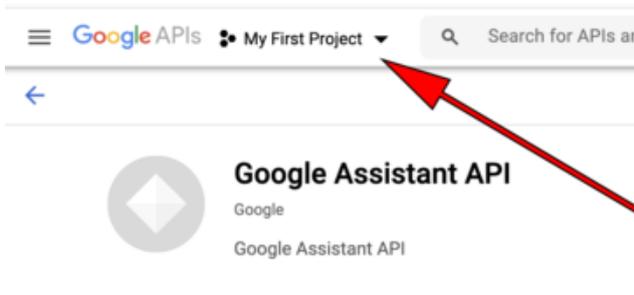


Under Traits, click **All 7 traits** and then click **Save Traits**

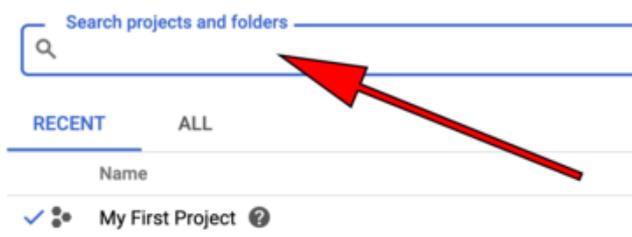
## Google Assistant API Setup

Go to the Google Developers Console to enable the API at <https://console.developers.google.com/apis/api/embeddedassistant.googleapis.com/overview> (<https://adafru.it/O8D>)

Before clicking the button, we need to select the project we created in the first step. By default, the incorrect project is selected.



Click on the project selector at the top.



This opens a dialog box of projects. If the project you created in the first step is not listed here, click inside the search box at the top.



Start typing the name of the project in the search box and click on the result that comes up. Click **Open**.



## Google Assistant API

Google

Google Assistant API

ENABLE

Now that the correct project is selected, we want to enable the Google API for this project. Click on the **Enable** button.

APIs & Services  
Google Assistant API

Overview  
Metrics  
Quotas  
Credentials

Credentials + CREATE CREDENTIALS DELETE

Credentials compatible with this API  
To view all credentials or create new credentials visit [Credentials in APIs & Services](#)

Remember to configure the OAuth consent screen with information about your application. [CONFIGURE CONSENT SCREEN](#)

OAuth 2.0 Client IDs

Name	Creation date
No OAuth clients to display	

Service Accounts

Email	Name	Usage with this service (last 30 days)
No service accounts to display		

+ CREATE CREDENTIALS DELETE

Compatible with this API  
To create new credentials visit [Credentials in APIs & Services](#)

Remember to configure the OAuth consent screen with information about your application. [CONFIGURE CONSENT SCREEN](#)

OAuth 2.0 Client IDs

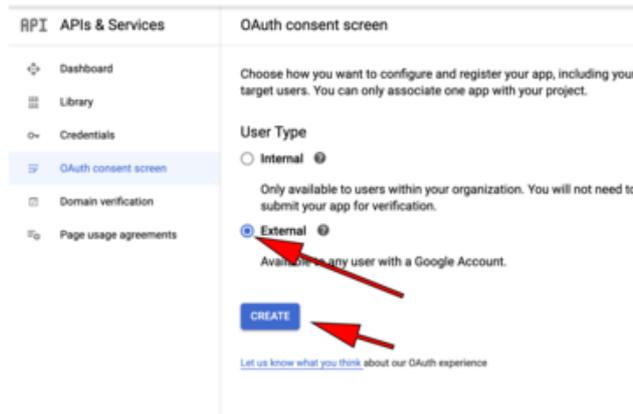
Creation date	Type	Client ID
No OAuth clients to display		

Service Accounts

[Manage service accounts](#)

Name	Usage with this service (last 30 days)	Usage with all services (last 30 days)
No service accounts to display		

Once it is enabled, you will be taken to the Overview Screen. Click on the **Credentials** tab on the left and then click the **Configure Consent Screen** button on the right.



For User Type, select **External** and Click **Create**



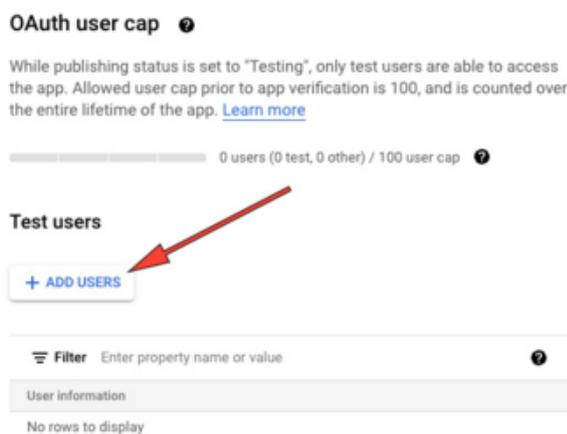
Enter an **Application Name**. This is the name that will appear on the permissions screen.

Select a **Support Email**. This is the email that will appear when you click on the Application name on the permissions screen.

You may need to add another **Email Address** under **Developer contact information** near the bottom of the form.

Scroll to the bottom and Click **Save and Continue**

## Add Test Users

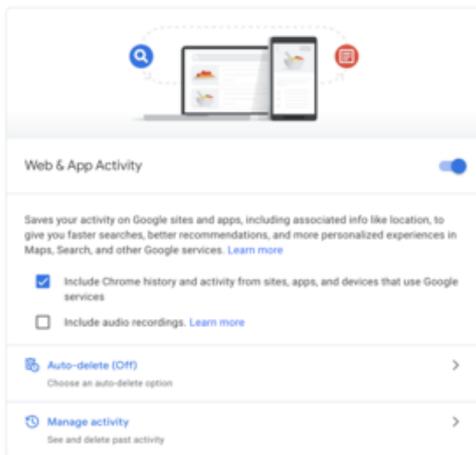


Next you'll need to add yourself as a test user to your app or you will get an Access Denied error when you try to authorize your credentials.



Type in the email addresses of any of your Google Accounts that you will be using to authorize the app. Then click **Save**.

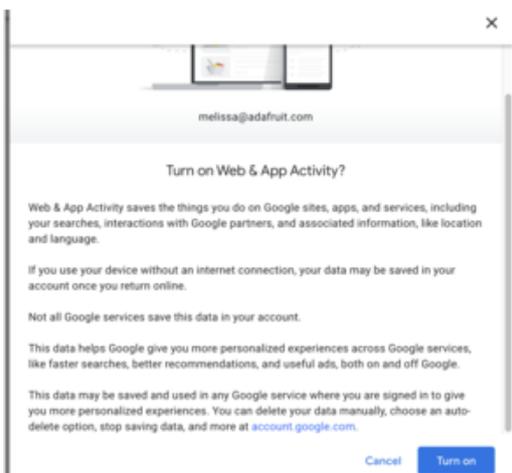
## Enabling Permissions



Go to Activity Controls at <https://myaccount.google.com/activitycontrols> (<https://adafruit.it/O8E>)

Make sure **Web & App Activity** is on.

Also make sure to select the **Include Chrome history and activity from sites, apps, and devices that use Google services** checkbox.



An additional dialog may pop up asking you to confirm. Click the **Turn on** button.

You may want to turn on **Location History** and **YouTube History** as well.

---

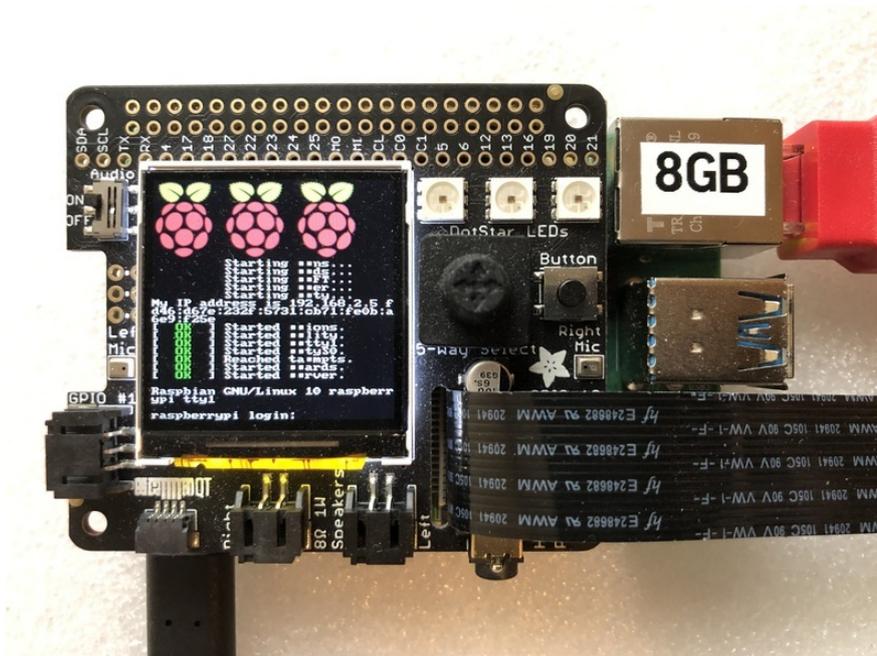
# Device Setup

Make sure you are using the 32-bit Lite version of Raspberry Pi OS. The desktop version has had some issues with Google Voice and the audio driver and Google drivers aren't available for the 64-bit version.

Next you'll want to setup all of the packages on the Raspberry Pi. If you haven't done so already, take a look at our [Adafruit BrainCraft HAT - Easy Machine Learning for Raspberry Pi](https://adafru.it/NLE) (<https://adafru.it/NLE>) guide if you are using the BrainCraft HAT.

If you are using the Voice Bonnet, you'll want to follow the setup steps in our [Adafruit Voice Bonnet](https://adafru.it/OjA) (<https://adafru.it/OjA>) guide.

These guides will take you through all the steps needed to get the Raspberry Pi updated and the **BrainCraft HAT** or **Voice Bonnet** all set up to the point we need to continue. In either case, make sure to install the 32-bit Lite version of Raspberry Pi OS.



Upload the `client-secret.json` OAuth 2.0 Credentials file that you downloaded in the Google Setup section to your Raspberry Pi and place it in your home directory. We're going to assume this is located at `/home/pi`. If it differs, please change any commands accordingly.

Next we're going to check that we have the required packages and setup a Virtual Environment.

```
cd ~
sudo apt-get update
sudo apt-get install -y python3-dev python3-venv pulseaudio
python3 -m venv env
env/bin/python -m pip install --upgrade pip setuptools wheel
source env/bin/activate
```

```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ sudo apt-get update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
Reading package lists... Done
pi@raspberrypi:~ $ sudo apt-get install python3-dev python3-venv
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-dev is already the newest version (3.7.3-1).
python3-dev set to manually installed.
The following NEW packages will be installed:
  python3-venv python3.7-venv
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,328 B of archives.
After this operation, 44.0 kB of additional disk space will be used.
Get:2 http://raspbian.raspberrypi.org/raspbian buster/main armhf python3-venv armhf 3
.7.3-1 [1,180 B]
Get:1 http://mirror.sjc02.svwh.net/raspbian/raspbian buster/main armhf python3.7-venv
armhf 3.7.3-2+deb10u2 [6,148 B]
Fetched 7,328 B in 1s (5,316 B/s)
```

Once you are done with that, your prompt should see **(env)** to the left to indicate you are in a Virtual Environment.

```
Cache entry deserialization failed, entry ignored
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/44/a6/7fb6e8b3f4a6051e72e4e2218
889351f0ee484b9ee17e995f5ccff780300/setuptools-50.3.0-py3-none-any.whl (785kB)
  100% |#####| 788kB 419kB/s
Collecting wheel
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading https://files.pythonhosted.org/packages/a7/00/3df031b3ecd5444d572141321
537080b40c1c25e1caa3d86cdd12e5e919c/wheel-0.35.1-py2.py3-none-any.whl
Installing collected packages: pip, setuptools, wheel
Found existing installation: pip 18.1
Uninstalling pip-18.1:
  Successfully uninstalled pip-18.1
Found existing installation: setuptools 40.8.0
Uninstalling setuptools-40.8.0:
  Successfully uninstalled setuptools-40.8.0
Successfully installed pip-20.2.3 setuptools-50.3.0 wheel-0.35.1
pi@raspberrypi:~ $ source env/bin/activate
(env) pi@raspberrypi:~ $
```

A recent change to a Google SDK dependency requires rust to be installed and the version of rust included in apt-get is too low of a version to work. To install a more recent version, use the following commands:

```
curl --proto 'https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

After it finishes running, you'll need to add it to your path:

```
export PATH=$PATH:/home/pi/.cargo/bin
source "$HOME/.cargo/env"
```

Install the Authorization library:

```
pip3 install --upgrade google-auth-oauthlib[tool]
```

Install the SDK with:

```
sudo apt install -y portaudio19-dev libffi-dev libssl-dev
pip3 install --upgrade google-assistant-library
git clone https://github.com/adafruit/assistant-sdk-python.git
cd assistant-sdk-python/google-assistant-sdk/
pip install .
cd
```

Install Additional Libraries:

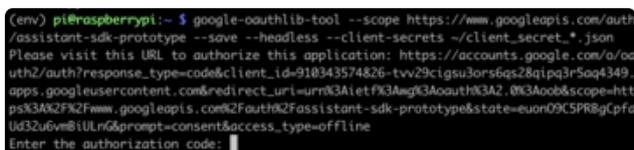
```
pip install --upgrade pyaudio sounddevice tenacity google-assistant-grpc google-api-
python-client adafruit-circuitpython-dotstar
```

## Generating an OAuth Token

Before we can run the script, we need to generate an OAuth Token. Run the following command:

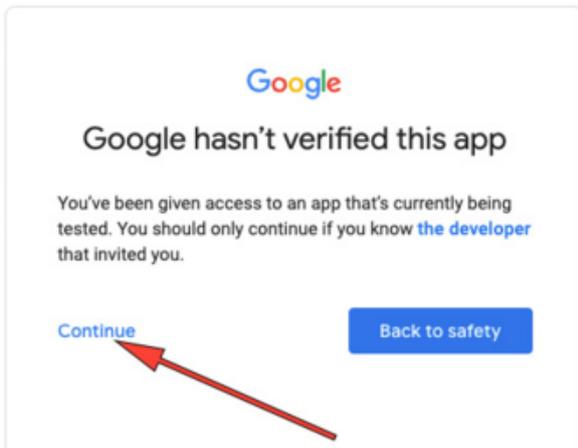
Be sure you have uploaded the `client_secret.json` file into your home directory or the OAuth Token generation script will fail.

```
google-oauthlib-tool --scope https://www.googleapis.com/auth/assistant-sdk-
prototype --save --headless --client-secrets ~/client_secret.json
```

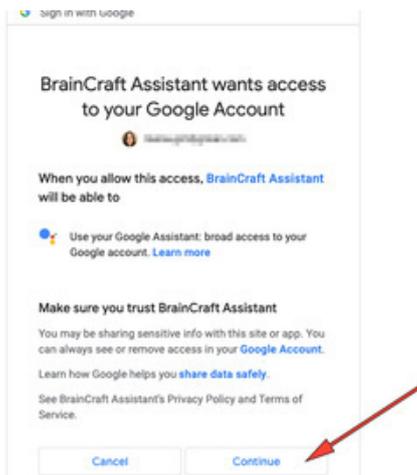


```
(env) pi@raspberrypi:~$ google-oauthlib-tool --scope https://www.googleapis.com/auth
/assistant-sdk-prototype --save --headless --client-secrets ~/client_secret.*.json
Please visit this URL to authorize this application: https://accounts.google.com/a/oa
uth2/auth?response_type=code&client_id=910343574826-tvw29cigsu3ors6qs28aipqr5oq4349.
apps.googleusercontent.com&redirect_uri=urn:ietf:params:oauth:state:oauth2_0n3AooB&scope=htt
ps://www.googleapis.com/auth/assistant-sdk-prototype&state=eun09C5PR8gCpfa
U632u6vm@iULnG&prompt=consent&access_type=offline
Enter the authorization code: █
```

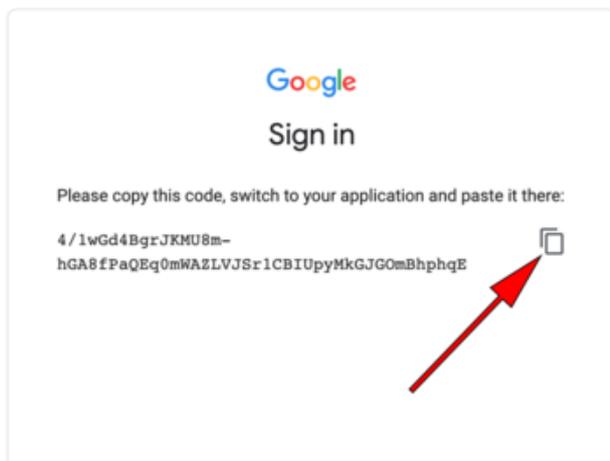
The script should provide a URL to visit to generate the token. Go ahead and copy and paste the URL into a web browser.



You may run into an alert that says your app isn't verified. Go ahead and click on the **Continue** link.



After all that, it will come up with a confirmation dialog with your account name and the permissions that you are granting. Go ahead and click on **Continue**.



Finally, you will be given an Authorization Code. Click on the **Copy** icon and it will get copied to your clipboard.

```
(env) pi@raspberrypi:~$ google-oauthlib-tool --scope https://www.googleapis.com
/auth/assistant-sdk-prototype --save --headless --client-secrets ~/client_secret
.json
Please visit this URL to authorize this application: https://accounts.google.com
/o/oauth2/auth?response_type=code&client_id=719705160719-ddv588vdse8re583qm9guc
d5pi3tero.apps.googleusercontent.com&redirect_uri=urn:ietf:params:oauth:response
_type:code&scope=https://www.googleapis.com/auth/assistant-sdk-prototype&state=
1Zd8tIxNgEUbxQ3x588Kqj19VnyX6&prompt=consent&access_type=offline
Enter the authorization code: 4/5QEwKj0yINGTRH2eGuSrRHY4FB5STb2kVJNuGacqJekgh_r
s0153YQ
credentials saved: /home/pi/.config/google-oauthlib-tool/credentials.json
(env) pi@raspberrypi:~$
```

Paste the Authorization Code back into the script and the token will be generated and saved.

Your Pi should now be all ready to go.

## Usage

Now that you have everything set up, it's time to use the Google Voice Assistant. Let's put a script on the Pi, run it and look at some of the things it can do.

First make sure you are in the Virtual Environment. If you have exited for some reason such as restarting the Pi, you can get back in by typing:

```
source env/bin/activate
```

## Running the Google Assistant Script

We wrote a script that will make use of the on-board DotStar LEDs and button on the BrainCraft HAT to make it easier to use the Google Assistant. Below is the full script. You can either save it onto your Pi by copying it below or you can type:

```
wget https://raw.githubusercontent.com/adafruit/Adafruit_Learning_System_Guides/
master/BrainCraft_Google_Assistant/gv_buttonotalk.py
```

and it will save it directly to the Pi.

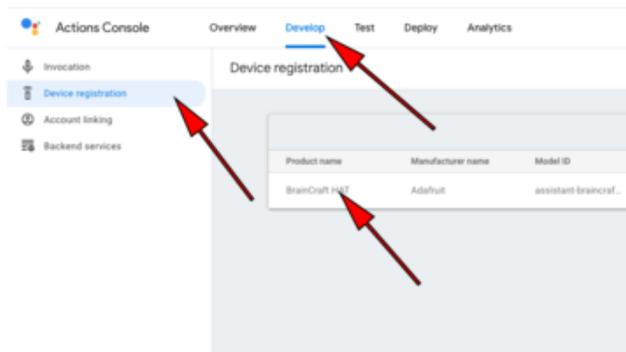
## Associating your Project ID and Model ID

Google recently made a change that requires you to run a sample script that will associate your Hardware Model ID and Project ID for the main script to work. **This step only needs to be done once.** To get your **Project ID** you can open your `client_secrets.json` file and find it there.

If you saved your **Model ID** during the Google Setup, you're all set. If not, you can find it by following these steps.



Open the Actions console at <https://console.actions.google.com/u/0/> (<https://adafruit.it/QAI>). Select your Project.



Click the **Develop** Tab at the top, then select **Device Registration**, then click on the product name.

Run the following command, swapping out [Your Project ID] and [Your Model ID] with the appropriate values:

```
googlesamples-assistant-pushtotalk --project-id [Your Project ID] --device-model-id [Your Model ID]
```

Press Control-C a couple of times to exit the script. Now you're ready to run the project script.

## Running the Project Script

To run the main script, type:

```
python3 gv_buttontotalk.py
```

Wait a few seconds until the script tells you to press a button and then press the button on the BrainCraft HAT. The lights should turn green, meaning it is waiting for you to give it a command or ask a question.



## Phrases to Try

Try asking some of these phrases:

What is the weather?

What movies are playing?

What is your favorite color?

What are the directions to the nearest grocery store?