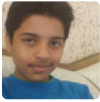




Google Assistant for PC/Mac/Linux



by justicesword11

Google Assistant is Google's answer to Amazon's Alexa smart home assistant. Initially only available with limited functionality in the Google Allo application, Google Assistant later rolled out with the Google Home and Pixel smartphones to bring the full power of Google's assistant to consumers.

Android 6.0+ also received Google Assistant, and just a few days ago Google launched the Google Assistant SDK which allows Assistant to be run on basically any platform. Today, we will show you how to install Google Assistant onto your Windows, Mac, or Linux machine using Python.

After a few months of waiting, smartphones running

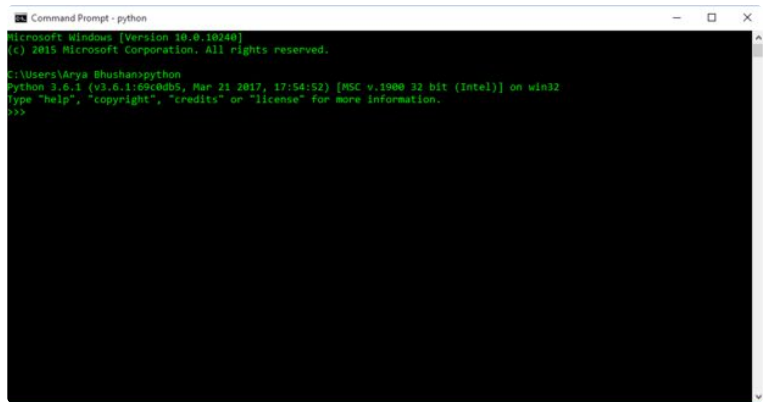
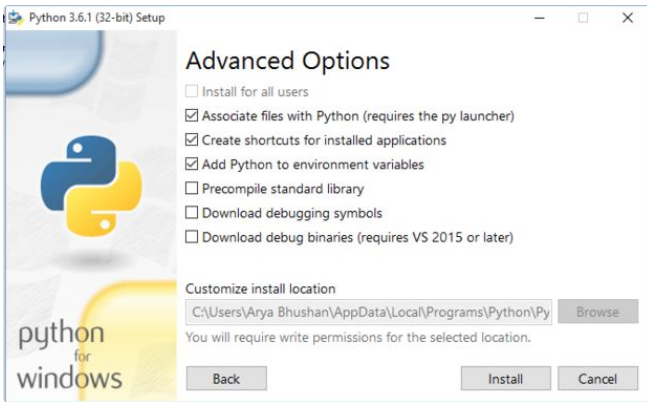


Step 1: Requirements

- Python 3

You'll need to have Python installed no matter whether or not you are using Windows, macOS, or a GNU/Linux distribution. Installation is fairly simple. Just open the installation file and select **customized installation**. In the next step click next, select the **Add Python to environment variables** checkbox and then move to next steps and install Python.

You can confirm Python is working by opening up a terminal/command prompt and then simply typing python. If you see the terminal/command prompt return the current Python version on your computer, then you're golden!



Step 2: Configure the Google Assistant API

What follows are step-by-step instructions walking you through the process to enable the Google Assistant API in the Cloud Platform Console so you can access Google Assistant through the Python program. All of these steps are platform independent, meaning that the steps are the same for Windows, macOS, and GNU/Linux OS.

1. Go to the Projects page in the Google Cloud Platform Console.
2. Click on “Create Project” up top.
3. Name the Project “My Google Assistant” and click “Create.”
4. Wait a few seconds for the Console to create your new Project. You should see a spinning progress icon in the top right. After it is done creating your Project, you will be brought to your Project’s configuration page.
5. Click this link to go straight to the Google Assistant API page. Up top, click “Enable.”
6. Google will warn you that you need to create credentials to use this API. Click “Create credentials” in the top right. This will take you to a setup wizard page where Google helps you figure out what kind of credentials you need to use this API.
7. Under “where will you be calling the API from”, select “Other UI (e.g. Windows, CLI tool)“. For “what data will you be accessing” select the “User data” circle. Now tap “what credentials do I need?”
8. Google should recommend that you create an OAuth 2.0 client ID. Name the Client ID anything you want, for example, your name + Desktop. Once done picking a name, click “create client ID.”
9. Under “product name shown to users” enter “My Google Assistant.” Click continue.
10. Click “done.” There’s no need to click download here as we only need the client secret, which we will download next.
11. Now under the list of OAuth 2.0 client IDs, you should see the client ID you just made. All the way to the right, click on the download icon to download the client_secret_XXX.json file, where ‘XXX’ is your client ID. Save this file anywhere on your computer, ideally in a new folder called “googleassistant.”
12. Go to the Activity controls page for your Google account and make sure that “Web & App Activity”, “Location History”, “Device Information”, and “Voice & Audio Activity” are enabled. This is so Google Assistant can actually read you personalized information.

We have now created a mechanism for a client, in this case our Windows/Mac/Linux machine, to access the Google Assistant API under our Google account. Next we need to set up the client that will access the Google Assistant API.

Step 3: Install the Google Assistant Sample Python Project

Open up a terminal/command prompt window and go through the following steps. First, enter this command:

py -m pip install google-assistant-sdk[samples]

You should see a whole bunch of dependencies being downloaded and installed when you enter this command. These are needed for the sample Python project to work. Wait for it to finish.

Once it's done, next enter the following command (**make sure to adjust the path**):

**pip install --upgrade google-auth-oauthlib[tool]
google-oauthlib-tool --client-secrets
path/to/client_secret_XXXXX.json --scope
https://www.googleapis.com/auth/assistant-sdk-
prototype --save --headless**

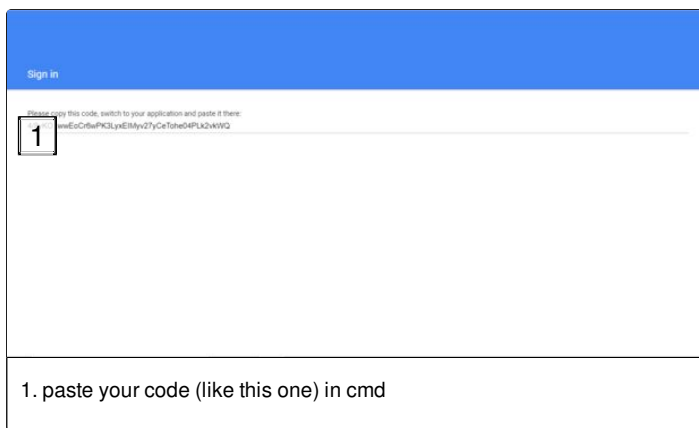
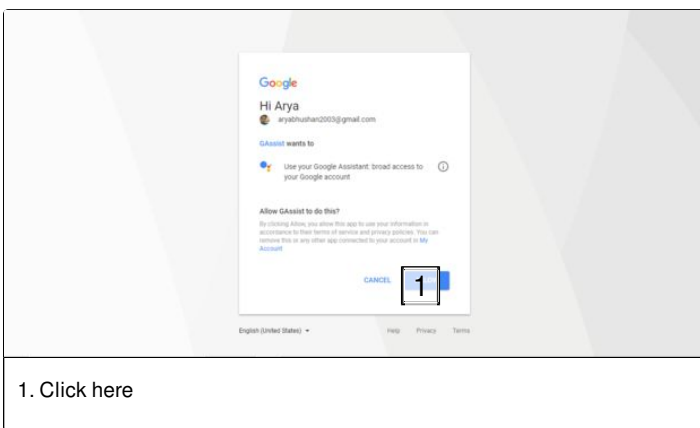
(As in my case, it was: `pip install --upgrade google-auth-oauthlib[tool]
google-oauthlib-tool --client-secrets "C:\Users\Arya Bhushan\Documents\GAssistant\additional\client_id.json" --scope
https://www.googleapis.com/auth/assistant-sdk-prototype --save --headless`)

In the command prompt, you will see a response telling you to visit a URL in order to authorize the application.

Copy and paste this URL into your browser. Select the same Google account that you used to configure the Google Assistant API. On the next page, you will see a text box that contains your client's Access Token.

Copy that Access token and paste it into the command prompt where it asks you for the authorization code. If done correctly, you will see a response that your credentials have been saved.

```
C:\Users\Arya Bhushan>py -m pip install google-assistant-sdk[samples]
```



```
credentials saved: C:\Users\Arya Bhushan\AppData\Roaming\google-oauthlib-tool\credentials.json
```

Step 4: Test the Google Assistant

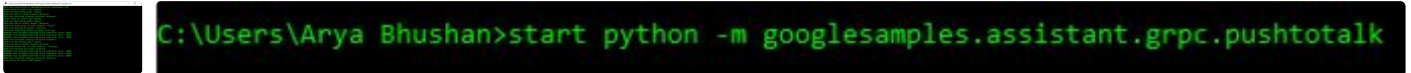
enter this command to begin talking to Google Assistant:

```
start python -m  
googlesamples.assistant.grpc.pushtotalk
```

Wait for the command prompt to say “press Enter to send a new request”, then press Enter to begin talking to Google Assistant. After you’re done speaking, the command prompt will display a transcript of what you just stated and then play back

the response. If you see a warning afterwards, just ignore it.

Have fun playing with Google Assistant on your Windows, macOS, or GNU/Linux machine! It’s not particularly useful in this format, but it is a very quick demonstration of the possibilities that the new Google Assistant SDK represents. Perhaps we may see desktop apps or browser extensions taking advantage of this functionality in the near future.

A terminal window with a black background and green text. The prompt is 'C:\Users\Arya Bhushan>' followed by the command 'start python -m googlesamples.assistant.grpc.pushtotalk'.

```
C:\Users\Arya Bhushan>start python -m googlesamples.assistant.grpc.pushtotalk
```

Step 5: Additional Info

Well this is my first instructables and if there are any mistakes or problem, feel free to comment and I will try to improve it!

Source: XDA

