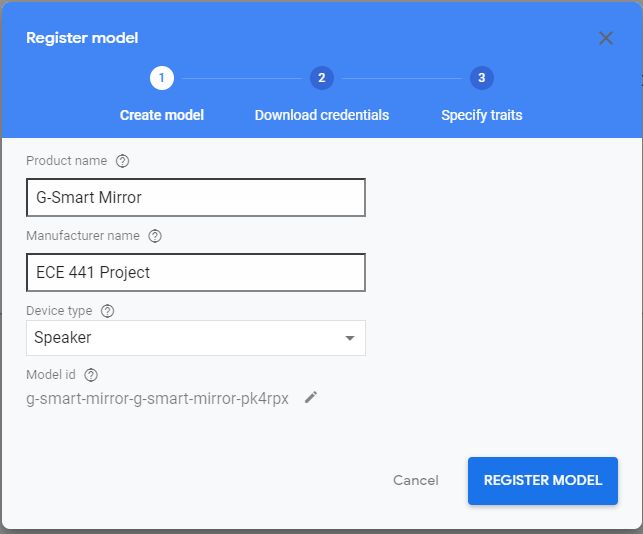
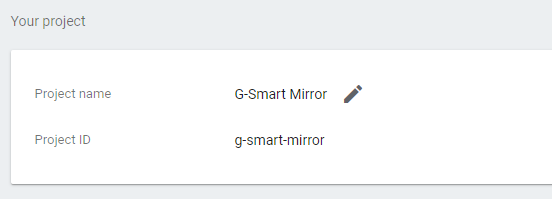
Google API Set-Up

Graphical user interface, text, application, Teams

Description automatically generated





Graphical user interface, application

Description automatically generated

Graphical user interface, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Registering for the Google API

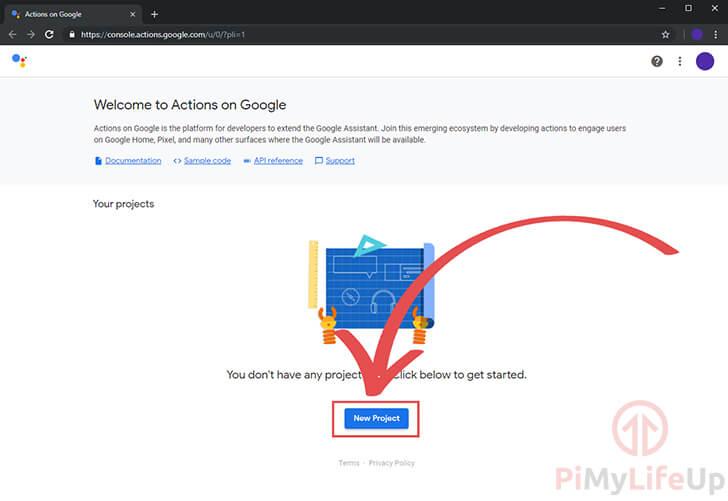
**1.** Before we get started with setting up the Google Assistant code on the Raspberry Pi itself, we must first register and set up a project on the Google Actions Console.

With your Google account ready to go to the [Google Console Actions dashboard](https://console.actions.google.com/), or just go to the URL below which will take you there.

https://console.actions.google.com

**2.** Once you have logged into your account, you will be greeted with the following screen.

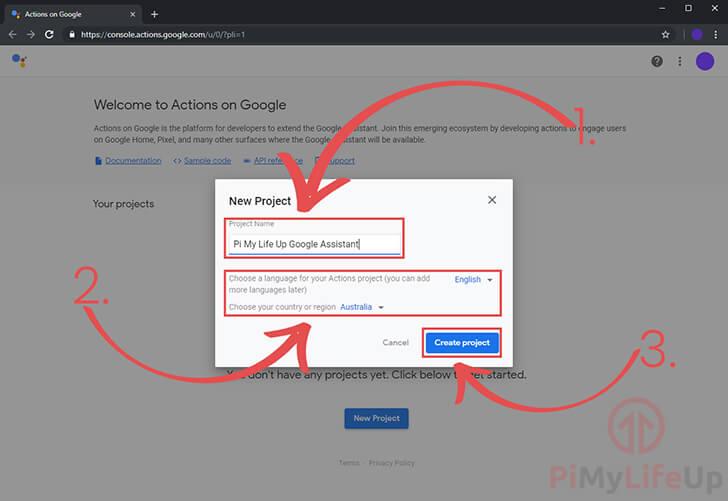
On here you will want to click the “**Add/Import project**” button as shown in our screenshot below.



**3.** On this next screen, you will be asked to enter a “**Project Name**” (**1.**)

In addition to a project name you need to set both your country and your language as shown in the screenshot (**2.**)

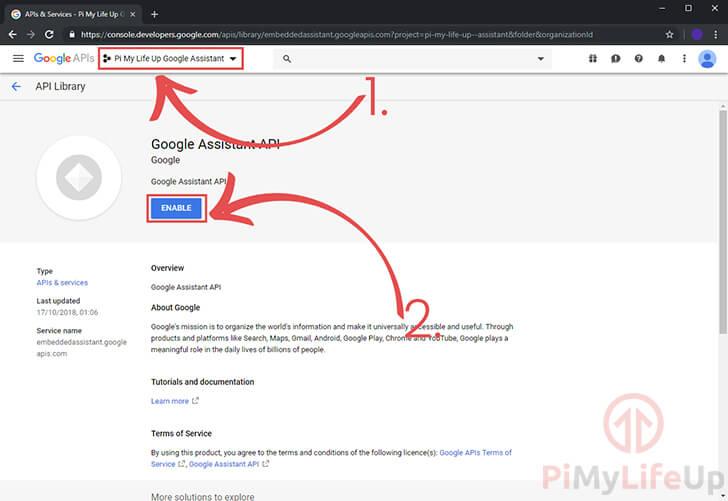
Once you have set the Project Name and chosen your language and country, click the “**Create Project**” (**3.**) button.



**4.** In a **new** tab, go to the [Google developers console](https://console.developers.google.com/apis/library/embeddedassistant.googleapis.com) and enable the Google Embedded Assistant API.

Now before you go ahead and click the “**Enable**” button make sure that you have your project selected (**1.**)

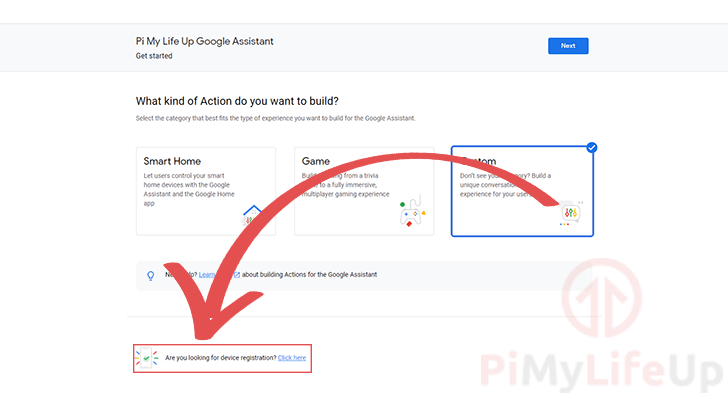
Once you are sure you have your current project selected, click the “**Enable**” button (**2.**)



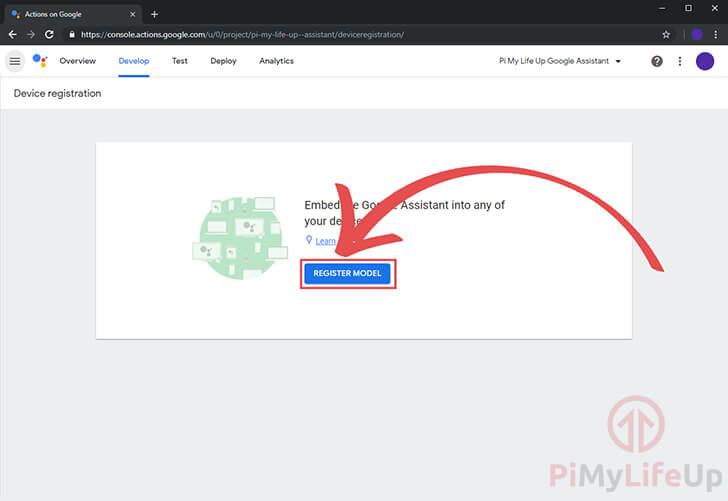
**5.** Now back in the other tab where you created the project, scroll down to the bottom of the screen.

You should see some text saying, “**Are you looking for device registration? Click here**“.

All you need to do to proceed is to click the “**click here**” text.



**6.** You will now be taken to the following screen, click the “**Register Model**” button to continue.



**7.** On this screen, you need to set a “**Product Name**“, “**Manufacturer name**” and set a “**Device Type**” (**1.**)

Below you can see the data that we entered into it, it doesn’t hugely matter what you set here, but all three boxes do need to be set for you to be able to register your model.

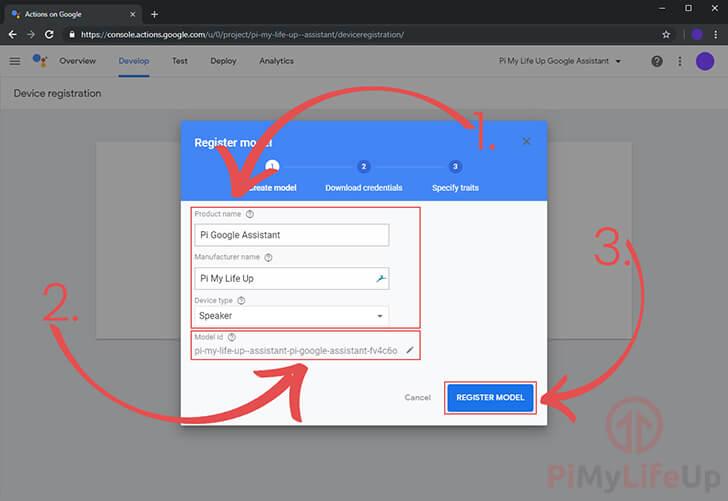
For the “**Product Name**” we just set this as a simple descriptor of what we are using this for, which in this tutorials case is simply a “**pi3 Google Assistant**“.

“**Manufacturer name**” doesn’t hugely matter as we have no intention of this being a widely used device, so we just set this to our websites name “**Pi My Life Up**“.

Lastly, we set the “**Device Type**” as “**Speaker**” as we felt it matched best what we intend on using the Google Assistant API for on our Raspberry Pi.

Make sure you write down then “**Device Model ID**” (**2.**) as you will need this later in the tutorial.

Finally, once everything is set, and you have written down the “**Device Model ID**” click the “**Register Model**” (**3.**) button to continue.



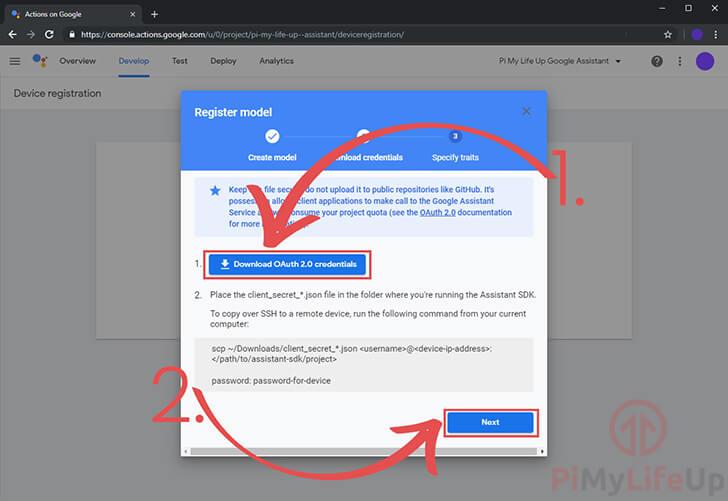
**8.** Now that you have registered the model you will now be taken to the “**Download credentials**” screen.

This screen is crucial as the provided credentials file is what we need for our Raspberry Pi 3 based Google Assistant to talk with the server.

To get this credentials file click the “**Download OAuth 2.0 credentials**” (**1.**) button as shown on the screenshot below.

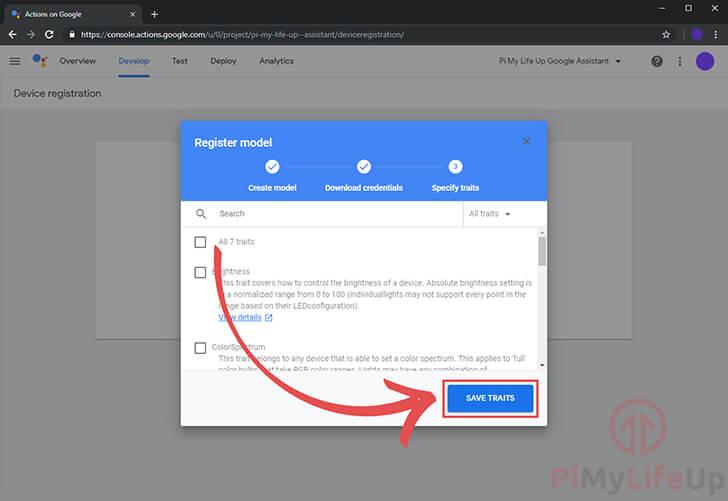
Keep this somewhere safe, as we will the text inside the file to the Raspberry Pi. (Of course, unless you downloaded it directly to your Pi)

Once you have the credentials safely stored on your on your computer or Raspberry Pi, you need to click the “**Next**” (**2.**) button.



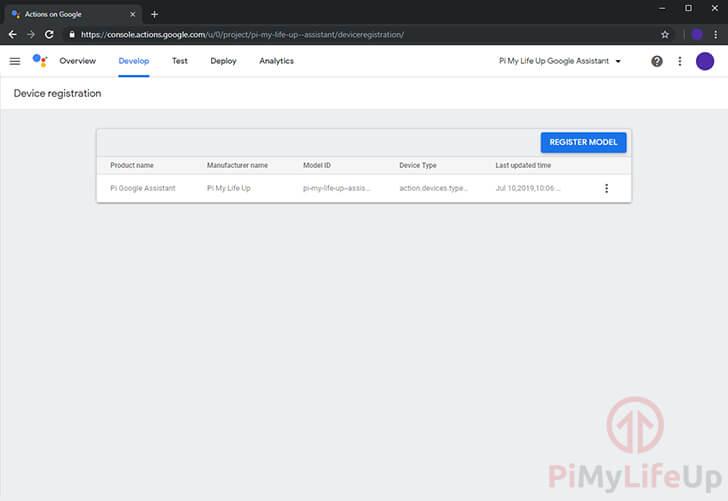
**9.** Finally, you can specify any traits that you might need.

In our case we don’t need any of these so we just clicked the “**Save Traits**” button as shown below.



**10.** Once everything is done, you should be shown on this screen.

We now only have one last thing we need to do before we can set up the Google Assistant on the Raspberry Pi itself.



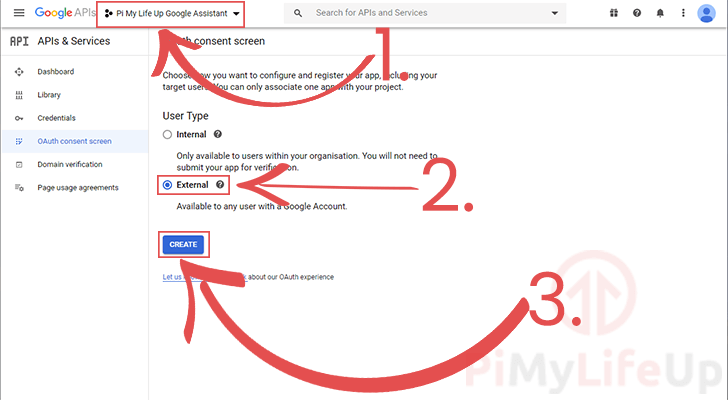
**11.** We also need to configure the OAuth consent screen. Without this, Google won’t let us authorize our Raspberry Pi Google Assistant device later in this tutorial.

To do this, you will need to go to the [API Credentials OAuth consent screen](https://console.developers.google.com/apis/credentials/consent) settings page.

On this first page, make sure that your project is selected in the top dropdown box (**1.**)

Secondly, you will need to select “**External**” as the user type (**2.**). This option will allow anyone with a Google account to utilize the Google Assistant software that we are setting up.

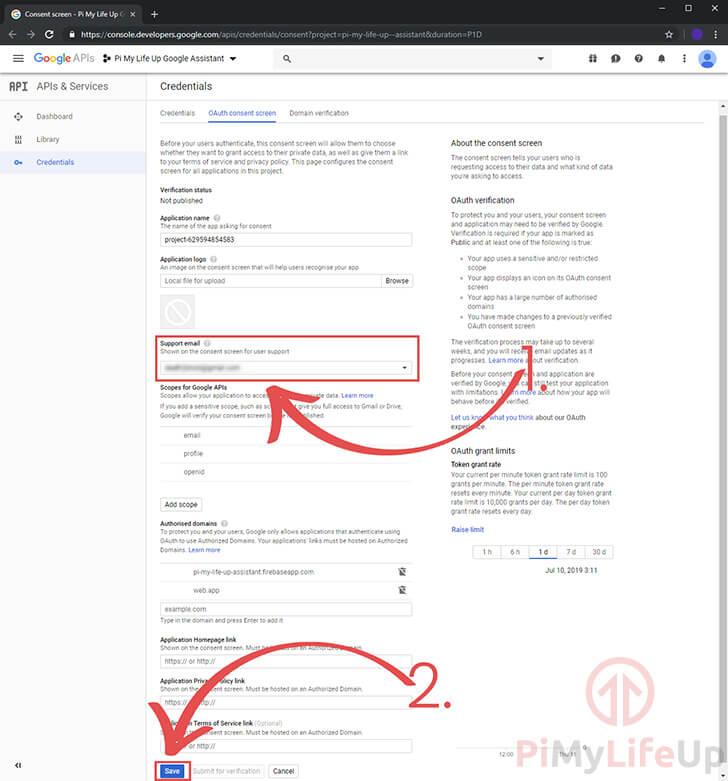
Once all configured, click the “**CREATE**” button to continue on to setting up the auth screen (**3.**).



**12.** This page will allow you to set up the details of the auth screen. As we don’t intend this to be publicly accessible there is only one option we are required to set.

Click the email dropdown to select your accounts email address (**1.**), selecting this will allow us to continue with the guide.

Once everything required has been set, click the “**Save**” button (**2.**) at the bottom of the page.



**13.** Finally, we need to go to the [Google My Account activity controls](https://myaccount.google.com/activitycontrols).

On here you will need to activate the following activity controls to ensure that the Google Assistant API works correctly.

* Web & App Activity
* Location History
* Device Information
* Voice & Audio Activity