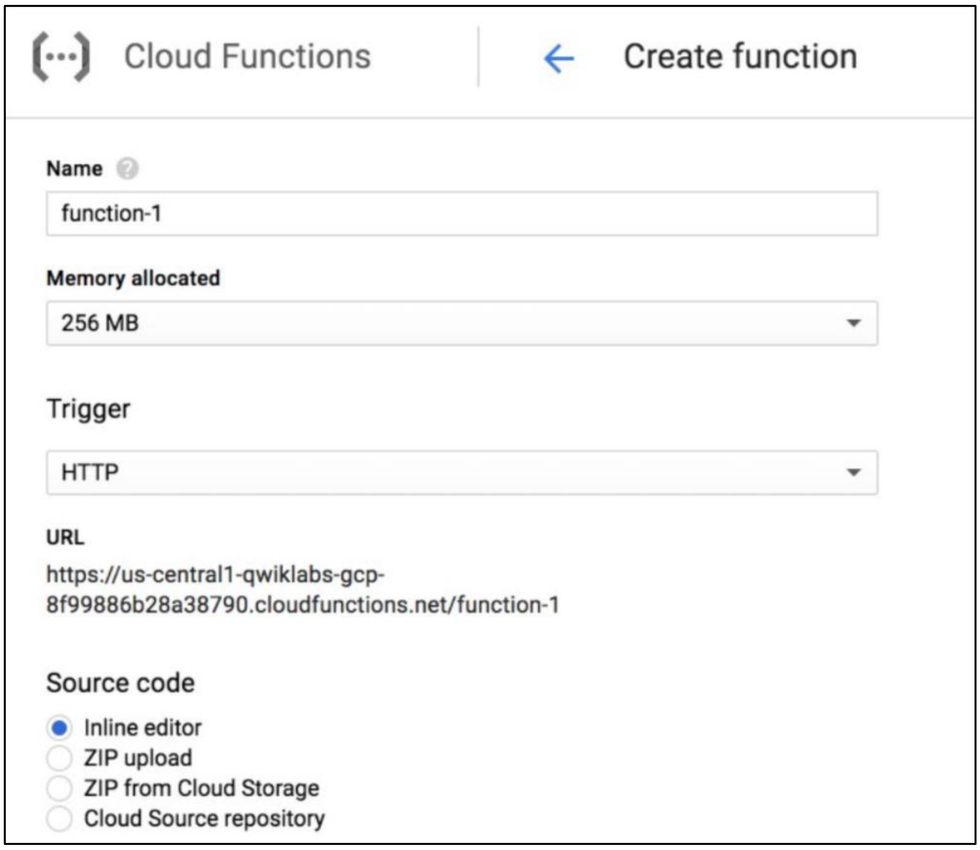
Build Interactive Apps with Google Assistant: Challenge Lab

Author: Vedant Kakde | **GitHub Profile:** github.com/vedant-kakde | **LinkedIn Profile:** linkedin.com/in/vedant-kakde/

Task 1: Initialize and configure a Cloud Function

Open the Navigation menu and select **Cloud Functions**, which is located under the compute header. Then click **Create function**.

This will open a template to create a new Cloud Function. Your page will resemble the following:



Name: function-1

Memory allocated: 256 MB

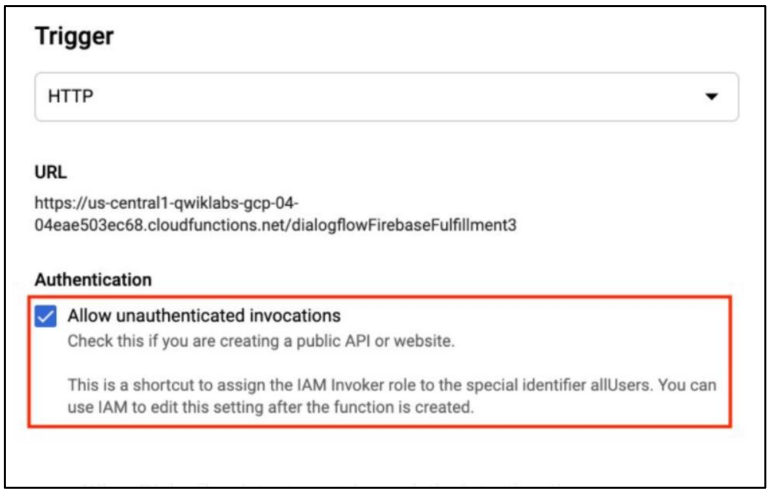
Trigger: HTTP

URL: https://us-central1-qwiklabs-gcp-8f99886b28a38790.cloudfunctions.net/function-1

Source code
- Inline editor (selected)
- ZIP upload
- ZIP from Cloud Storage
- Cloud Source repository

For the Cloud Function's **Name** field, enter in **magic_eight_ball**

Then scroll down to the authentication section and check the box next to "Allow unauthenticated invocations":



Trigger: HTTP

URL: https://us-central1-qwiklabs-gcp-04-04eae503ec68.cloudfunctions.net/dialogflowFirebaseFulfillment3

Authentication

✓ Allow unauthenticated invocations
Check this if you are creating a public API or website.

This is a shortcut to assign the IAM Invoker role to the special identifier allUsers. You can use IAM to edit this setting after the function is created.

Forgetting to do the above will cause your simulation test to fail at the end!

Click **SAVE**

Now click **NEXT** and find the inline editor for **MAIN.PY** and **REQUIREMENTS.TXT**. Make sure that the MAIN.PY tab is open. If not already present create them. This file defines your fulfillment logic and is used to create and deploy a Cloud Function. Here are some specifics on its basic functioning:

- When Dialogflow intents are triggered, the intent's action name (declared in the action area of the intent) is provided to you in the request to your fulfillment. You use this action name to determine what logic to carry out.
- Within every request to your fulfillment, if Dialogflow parsed parameters from the user input, you can access the parameter by name. Here, you declare the names of the parameters so you can access them later.

Now that you have a better understanding of MAIN.PY, you will build out the function's fulfillment logic. Remove the boilerplate code from the file. Then copy and paste the following code into **MAIN.PY**:

```
import random
import logging
import google.cloud.logging
from google.cloud import translate_v2 as translate
from flask import Flask, request, make_response, jsonify

def magic_eight_ball(request):

client = google.cloud.logging.Client()
client.get_default_handler()
client.setup_logging()

choices = [
"It is certain.", "It is decidedly so.", "Without a doubt.",
"Yes - definitely.", "You may rely on it.", "As I see it, yes.",
"Most likely.", "Outlook good.", "Yes.","Signs point to yes.",
"Reply hazy, try again.", "Ask again later.",
"Better not tell you now.", "Cannot predict now.",
"Concentrate and ask again.", "Don't count on it.",
"My reply is no.", "My sources say no.", "Outlook not so good.",
"Very doubtful."
]

magic_eight_ball_response = random.choice(choices)

logging.info(magic_eight_ball_response)

return make_response(jsonify({'fulfillmentText': magic_eight_ball_response }))
```

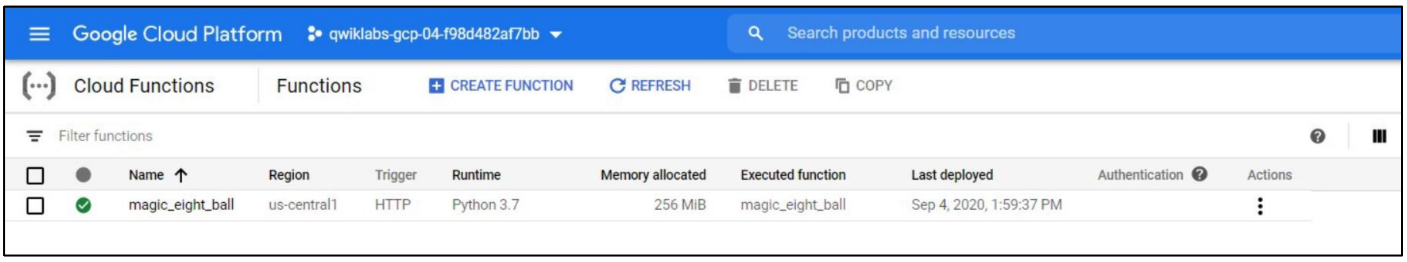
Now open the REQUIREMENTS.TXT. Replace the contents of the file with the following:

REQUIREMENTS.TXT

```
google-cloud-translate
google-cloud-logging
```

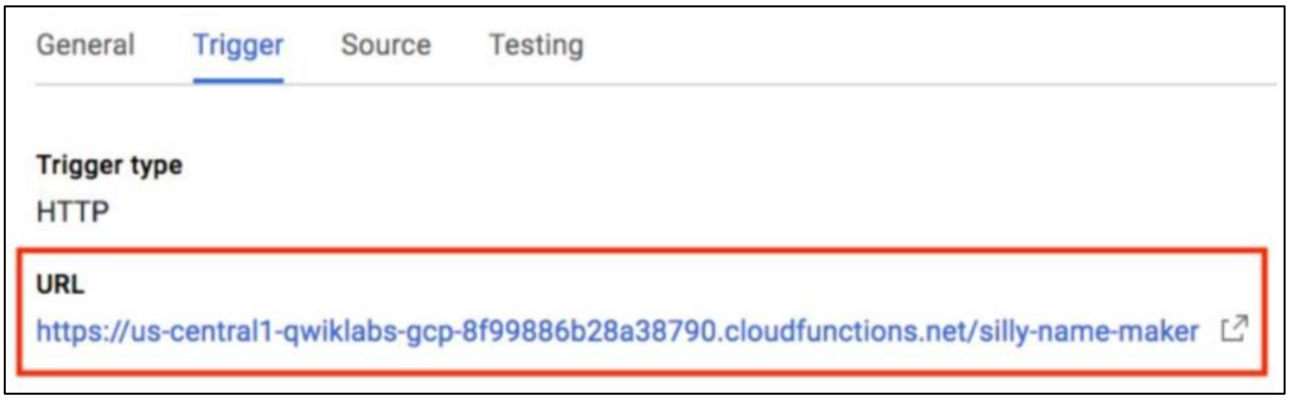
Once you have those files configured, now change the **runtime** to **Python 3.7.** Then find the **Entry point** field. Enter in **magic_eight_ball** for the value.

Now click the **Deploy** button below. It will take about a minute for your function to be built. When the creation completes, your overview page will resemble the following



		Name ↑	Region	Trigger	Runtime	Memory allocated	Executed function	Last deployed	Authentication	Actions
☐	✓	magic_eight_ball	us-central1	HTTP	Python 3.7	256 MiB	magic_eight_ball	Sep 4, 2020, 1:59:37 PM		⋮

Now click on the **magic_eight_ball** function to get more details about it's configuration. Then click on the **trigger** tab. You will see a function URL that resembles the following:



Trigger type
HTTP

URL
https://us-central1-qwiklabs-gcp-8f99886b28a38790.cloudfunctions.net/silly-name-maker

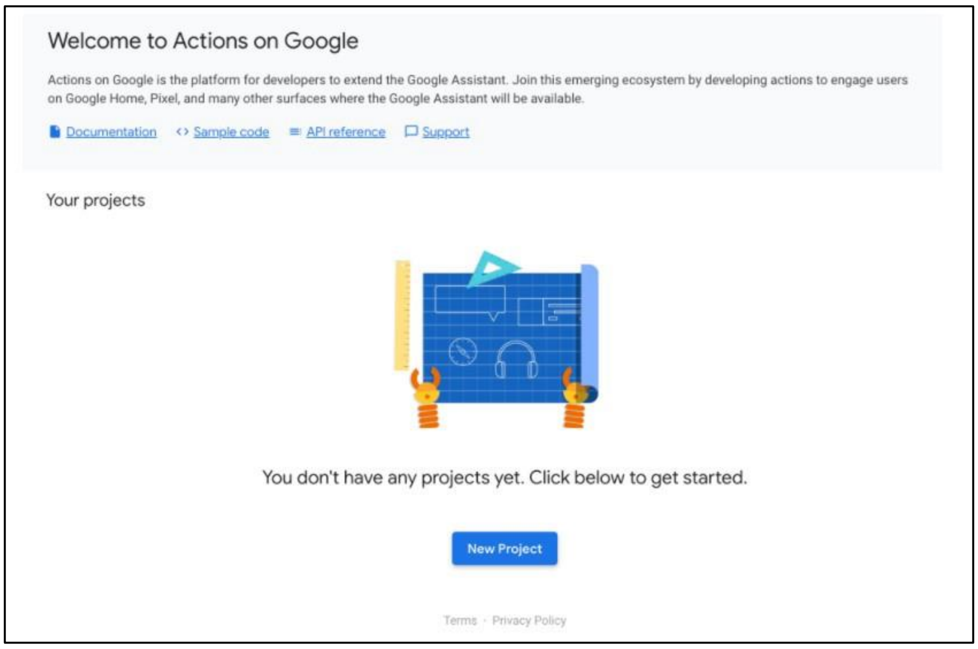
Copy the function URL in a text editor. You will use it as the URL for the Dialogflow webhook, which is configured in the next section.

Click **Check my progress** to verify your performed task.

Task 2: Create the Lab Magic 8 Ball app for Google Assistant

Regardless of the Assistant application you're building, you will always have to create an Actions project so your app has an underlying organizational unit.

Open the Actions on Google Developer Console(see lab manual for link) in a new tab. Sign in with your Qwiklabs credentials if prompted. You should be looking at a clean Actions console that resembles the following:



Welcome to Actions on Google

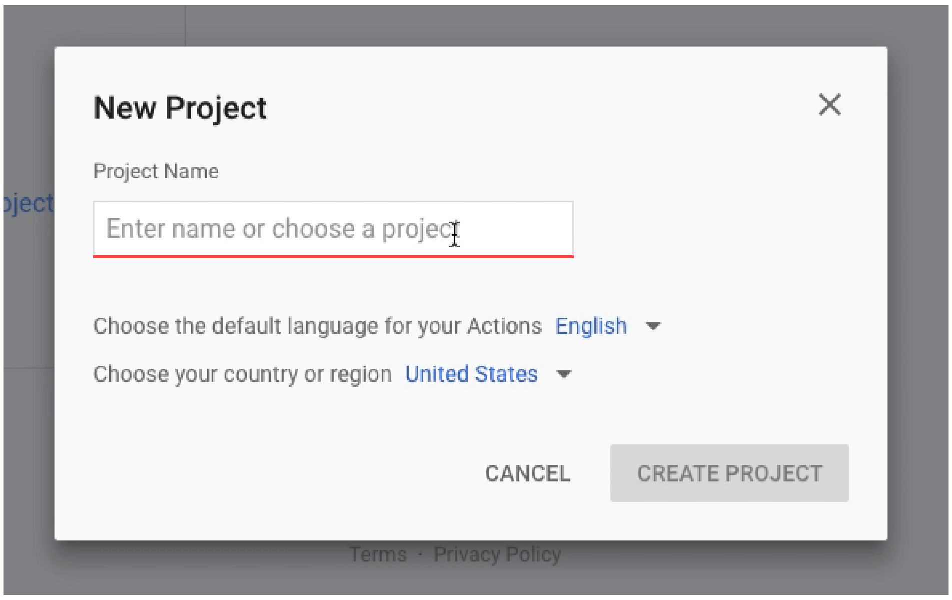
Actions on Google is the platform for developers to extend the Google Assistant. Join this emerging ecosystem by developing actions to engage users on Google Home, Pixel, and many other surfaces where the Google Assistant will be available.

Your projects

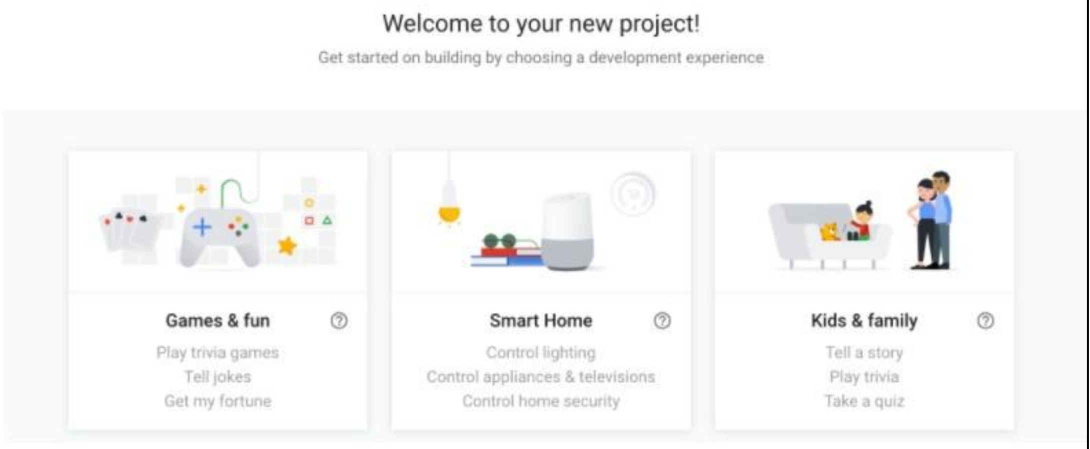
You don't have any projects yet. Click below to get started.

Click **New Project** and agree to Actions on Google's terms of service when prompted.

Click into the Project Name field and select your Qwiklabs Google Cloud project ID from the dropdown. Then click **Import project**:


New Project
Project Name
Enter name or choose a project
Choose the default language for your Actions English
Choose your country or region United States
CANCEL
CREATE PROJECT
Terms · Privacy Policy

Soon after you will be presented with a welcome page that resembles the following:



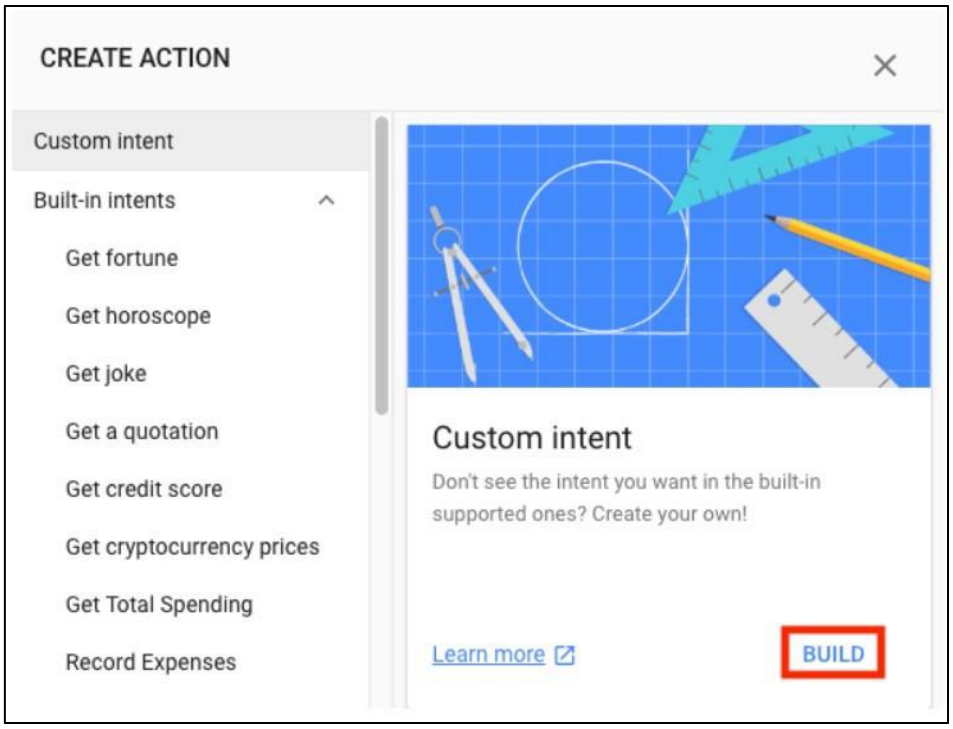
Now click **Actions Console** in the top left corner to return to the homepage. Then click on the project you just created (title has your Project ID as the name.)

Build an Action

An action is an interaction you build for the Google Assistant. An action supports a specific intent (a goal or task that users want to accomplish), which is carried out by a corresponding fulfillment (logic that handles an intent and carries out the corresponding Action.) You will now build an Action that supports silly name generation.

Click on your project name. Then from the center menu click **Build your Action** . Add a display name with your **initials + magic 8 ball.** Click **SAVE**

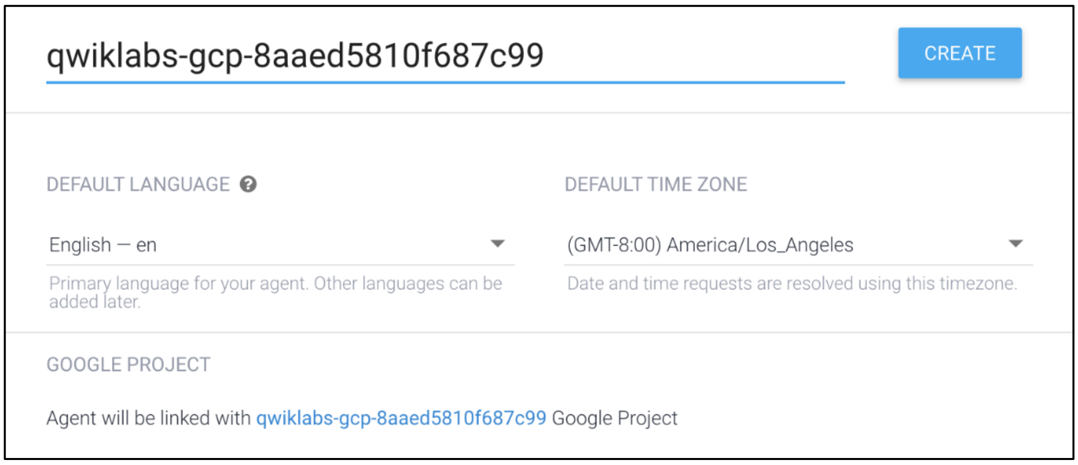
Now click Actions > **Get Started**. Then select **Custom Intent** > **BUILD**:



This will take you to the Dialogflow console. Select your Qwiklabs account and click **Allow** when Dialogflow prompts you for permission to access your Google Account.

When you land on the Dialogflow account settings page, check the box next to **Yes, I have read and accept the agreement** and click **Accept**.

If you are brought to the following Dialogflow agent creation page, click **CREATE:**


qwiklabs-gcp-8aaed5810f687c99

CREATE

DEFAULT LANGUAGE

English — en

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

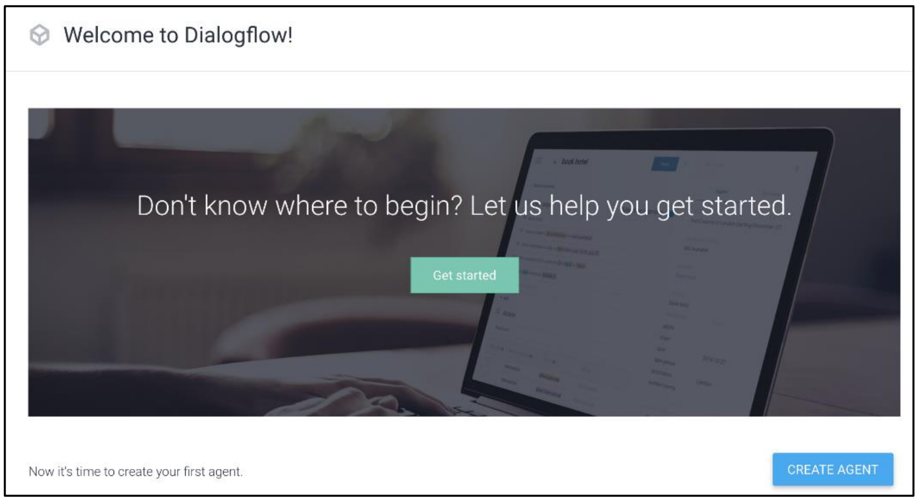
(GMT-8:00) America/Los_Angeles

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Agent will be linked with qwiklabs-gcp-8aaed5810f687c99 Google Project

If you are brought to this page instead:


Welcome to Dialogflow!

Don't know where to begin? Let us help you get started.

Get started

Now it's time to create your first agent.

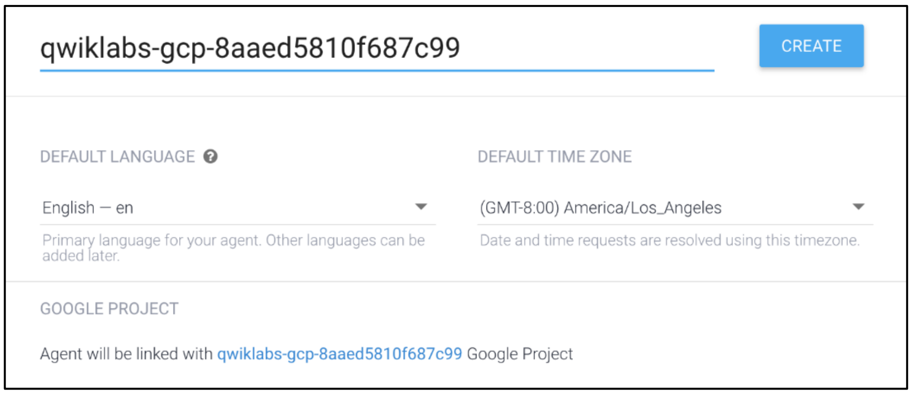
CREATE AGENT

Close the Dialogflow agent creation tab. You will return to the Actions Console.

Click **Get Started** > **Custom Intent** > **BUILD**.

Select your Qwiklabs account and click **Allow** when Dialogflow prompts you for permission to access your Google Account.

Now click **CREATE:**



qwiklabs-gcp-8aaed5810f687c99

DEFAULT LANGUAGE

English — en

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT-8:00) America/Los_Angeles

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

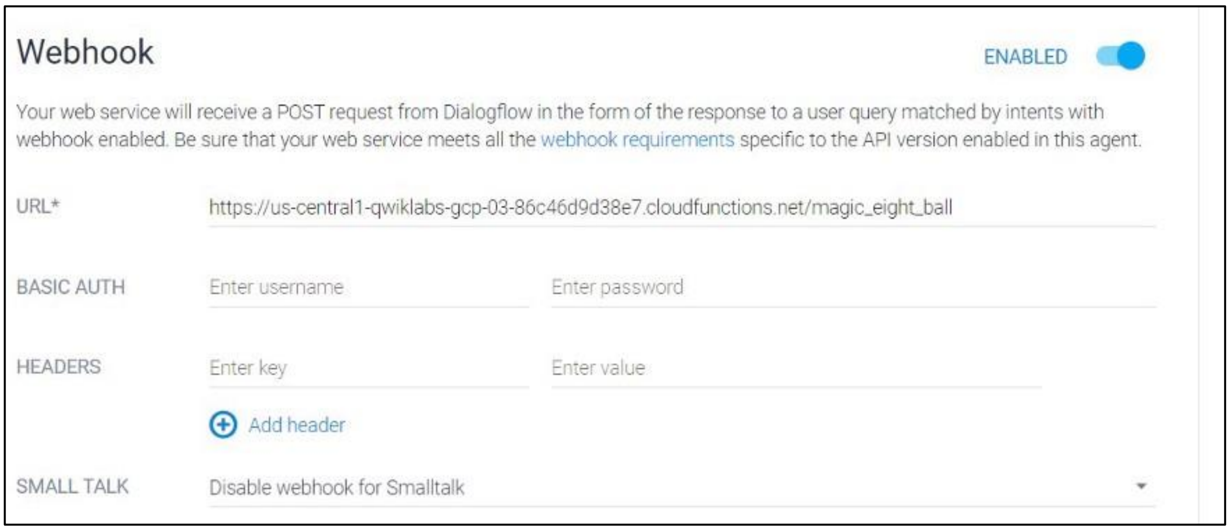
Agent will be linked with qwiklabs-gcp-8aaed5810f687c99 Google Project

An agent is an organizational unit that collects information needed to complete a user's request, which it then forwards to a service that provides fulfillment logic.

You will now build the basic framework for fulfillment logic.

Click **Fulfillment** from the left-hand menu. Move the slider for **Webhook** to the right, setting it to **Enabled**.

Copy and paste the **cloud function URL.** Your page should resemble the following:



Webhook

ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements specific to the API version enabled in this agent.

URL* https://us-central1-qwiklabs-gcp-03-86c46d9d38e7.cloudfunctions.net/magic_eight_ball

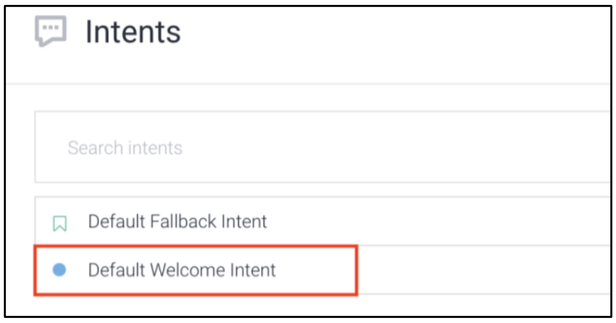
BASIC AUTH Enter username Enter password

HEADERS Enter key Enter value

SMALL TALK Disable webhook for Smalltalk

Click **Save**

Scroll down and click **Save** in the bottom right corner. Then click **Intents** from the left hand menu and select **Default Welcome Intent**:



Intents

Default Fallback Intent

Default Welcome Intent

Delete all text responses
Click **add response > text response**
Add this "Welcome to the lab magic 8 ball, ask me a yes or no question and I will predict the future!"

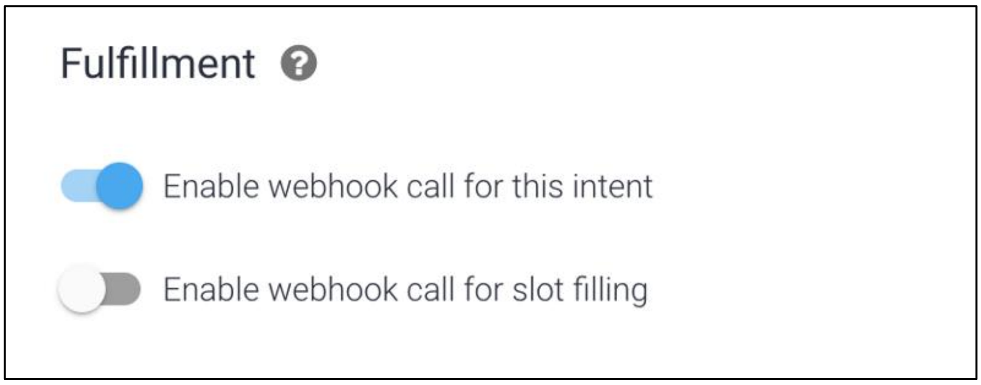
Click **Save**

Again click on **indents**

Click **default fallback intent**

Delete all the responses

Enable **Set this intent as end of conversation** and Now scroll down and expand the **Fulfillment** section and click **Enable fulfillment**. Then click the **Enable webhook call for this intent** slider:



Enable webhook call for this intent: on
Enable webhook call for slot filling: off

Click **Save.**

This tells Dialogflow to call your fulfillment to generate a response to the user instead of using Dialogflow's response feature.

Test your Assistant application with the Actions simulator

Now that you your Cloud Function has been deployed and your webhook has been properly set up, you can preview the app in the Actions simulator.

Check your Google permission settings

Switch to the **develop browser tab**

Click **test**

Click **visit activity controls**

Ensure that the following permissions are enabled by sliding the toggles to **TURN ON** the following cards:

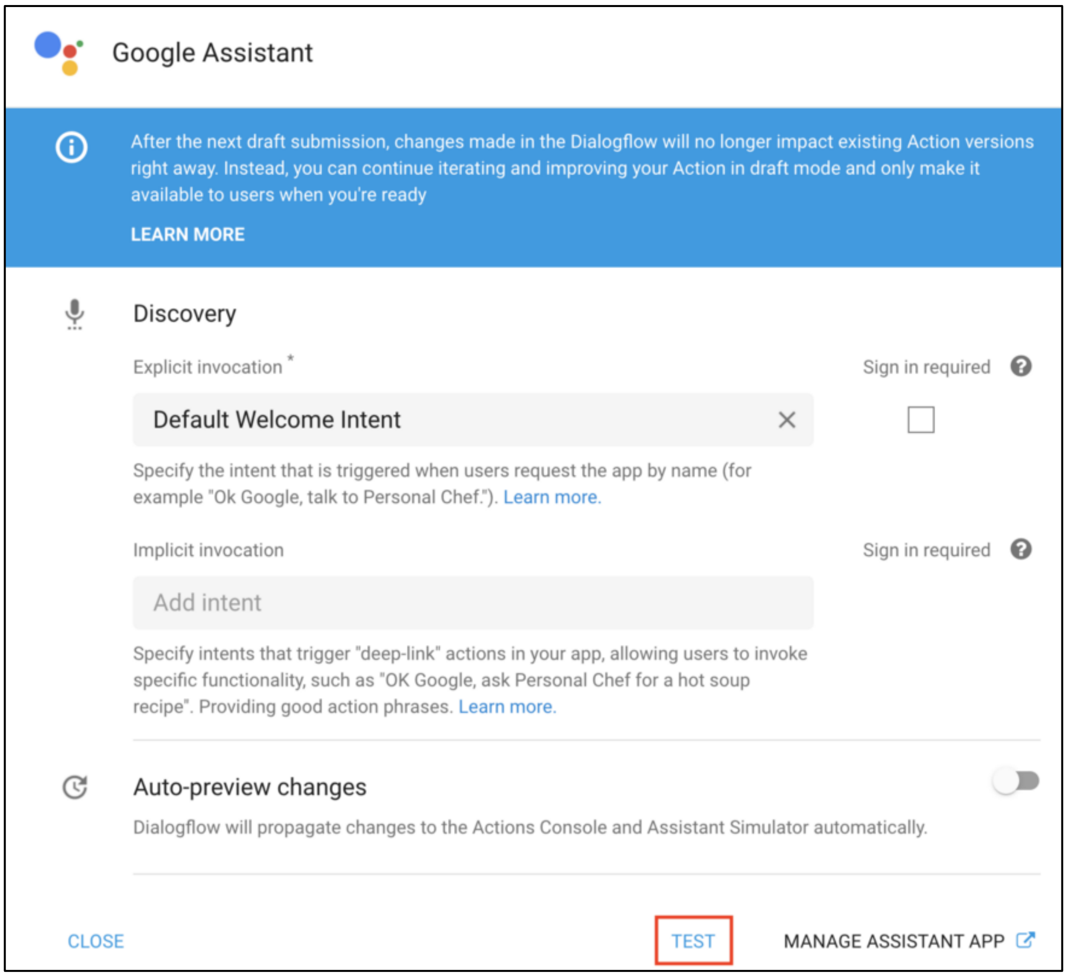
- Web & App Activity

Now **close** the Activity Controls page.

Test the application with the simulator

Return to the Dialogflow console. Then from the left-hand menu, click **Integrations**. Then select **Integration Settings**.

Once you land on the following page, click **TEST**:



Google Assistant

After the next draft submission, changes made in the Dialogflow will no longer impact existing Action versions right away. Instead, you can continue iterating and improving your Action in draft mode and only make it available to users when you're ready

LEARN MORE

Discovery

Explicit invocation *

Default Welcome Intent

Sign in required

Specify the intent that is triggered when users request the app by name (for example "Ok Google, talk to Personal Chef."). Learn more.

Implicit invocation

Add intent

Sign in required

Specify intents that trigger "deep-link" actions in your app, allowing users to invoke specific functionality, such as "OK Google, ask Personal Chef for a hot soup recipe". Providing good action phrases. Learn more.

Auto-preview changes

Dialogflow will propagate changes to the Actions Console and Assistant Simulator automatically.

CLOSE TEST MANAGE ASSISTANT APP

Now you should be on test browser tab. To invoke the Action, hit the enter key in the **Talk to my test app** box near the bottom of the simulator console. You should be presented with a similar response:

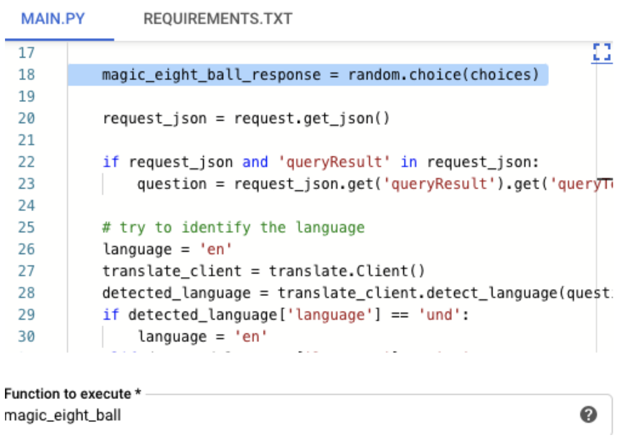
Now enter : Will I complete this challenge lab?

It will return a random reply.

Click **Check my progress** to verify your performed task.

Task 2: Add multilingual support to your magic_eight_ball Cloud Function

Switch to cloud function, edit the function, and add the following code to the line after : *magic_eight_ball_response = random.choice(choices)*



MAIN.PY REQUIREMENTS.TXT

```
    magic_eight_ball_response = random.choice(choices)

    request_json = request.get_json()

    if request_json and 'queryResult' in request_json:
        question = request_json.get('queryResult').get('queryT

    # try to identify the language
    language = 'en'
    translate_client = translate.Client()
    detected_language = translate_client.detect_language(quest
    if detected_language['language'] == 'und':
        language = 'en'
```

Function to execute *

magic_eight_ball

request_json = request.get_json() if request_json and ‘queryResult’ in request_json: question = request_json.get(‘queryResult’).get(‘queryText’) # *try to identify the language* language = ‘en’ translate_client = translate.Client() detected_language = translate_client.detect_language(question) if detected_language[‘language’] == ‘und’: language = ‘en’ elif detected_language[‘language’] != ‘en’: language = detected_language[‘language’] # *translate if not english* if language != ‘en’: logging.info(‘translating from en to %s’ % language) translated_text = translate_client.translate(magic_eight_ball_response, target_language=language) magic_eight_ball_response = translated_text[‘translatedText’]

Click **Deploy**

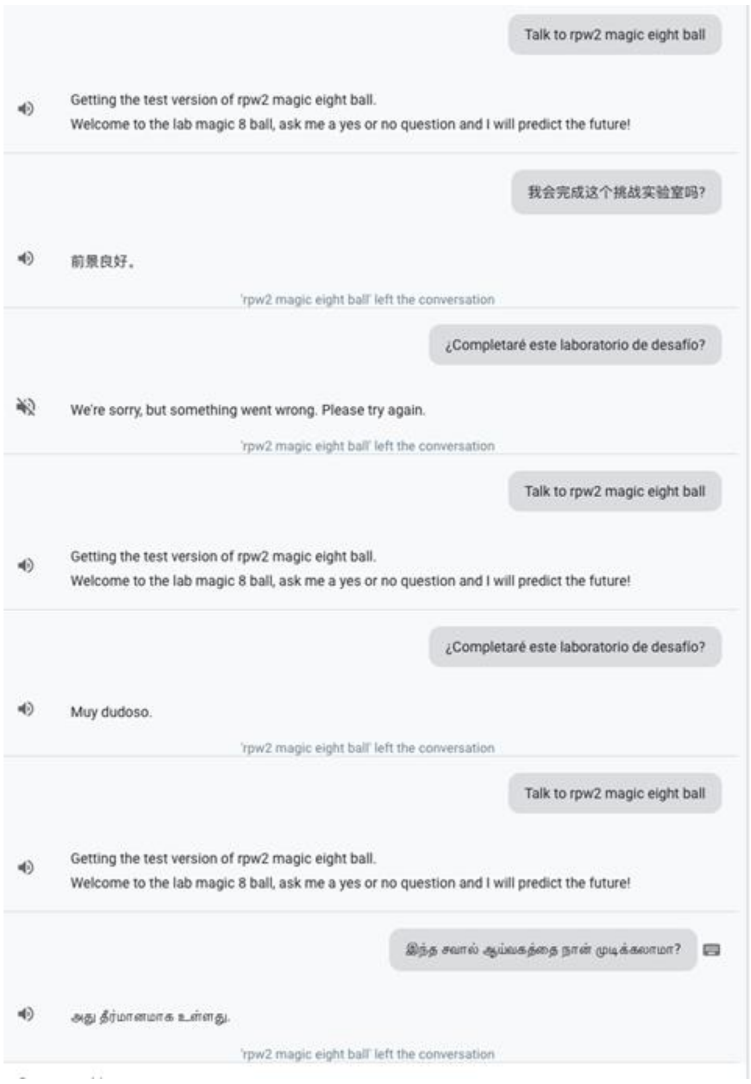
Switch to test and try the app again using the following lines

· 我会完成这个挑战实验室吗？

· ¿Completaré este laboratorio de desafío?

· இந்த சவால் ஆய்வகத்தை நான் முடிக்கலாமா?

Each line should reply in the same language



Talk to rpw2 magic eight ball

Getting the test version of rpw2 magic eight ball.
Welcome to the lab magic 8 ball, ask me a yes or no question and I will predict the future!

我会完成这个挑战实验室吗?

前景良好。

'rpw2 magic eight ball' left the conversation

¿Completaré este laboratorio de desafío?

We're sorry, but something went wrong. Please try again.

'rpw2 magic eight ball' left the conversation

Talk to rpw2 magic eight ball

Getting the test version of rpw2 magic eight ball.
Welcome to the lab magic 8 ball, ask me a yes or no question and I will predict the future!

¿Completaré este laboratorio de desafío?

Muy dudoso.

'rpw2 magic eight ball' left the conversation

Talk to rpw2 magic eight ball

Getting the test version of rpw2 magic eight ball.
Welcome to the lab magic 8 ball, ask me a yes or no question and I will predict the future!

இந்த சவால் ஆய்வகத்தை நான் முடிக்கலாமா?

அது தீர்மானமாக உள்ளது.

'rpw2 magic eight ball' left the conversation

Click **Check my progress** to verify your performed task.

Congratulations! You completed this challenge lab.