PRZEDMIOT: Systemy dialogowe

LABORATORIUM nr 5

Chatbot is a program that can conduct an intelligent conversation. It should be able to convincingly simulate a human behaviour and pass the turing test.

In this series we will learn how to build a chatbot from scratch, connect it with external API to pull data and deploy it on Slack as a bot.

Series Content:

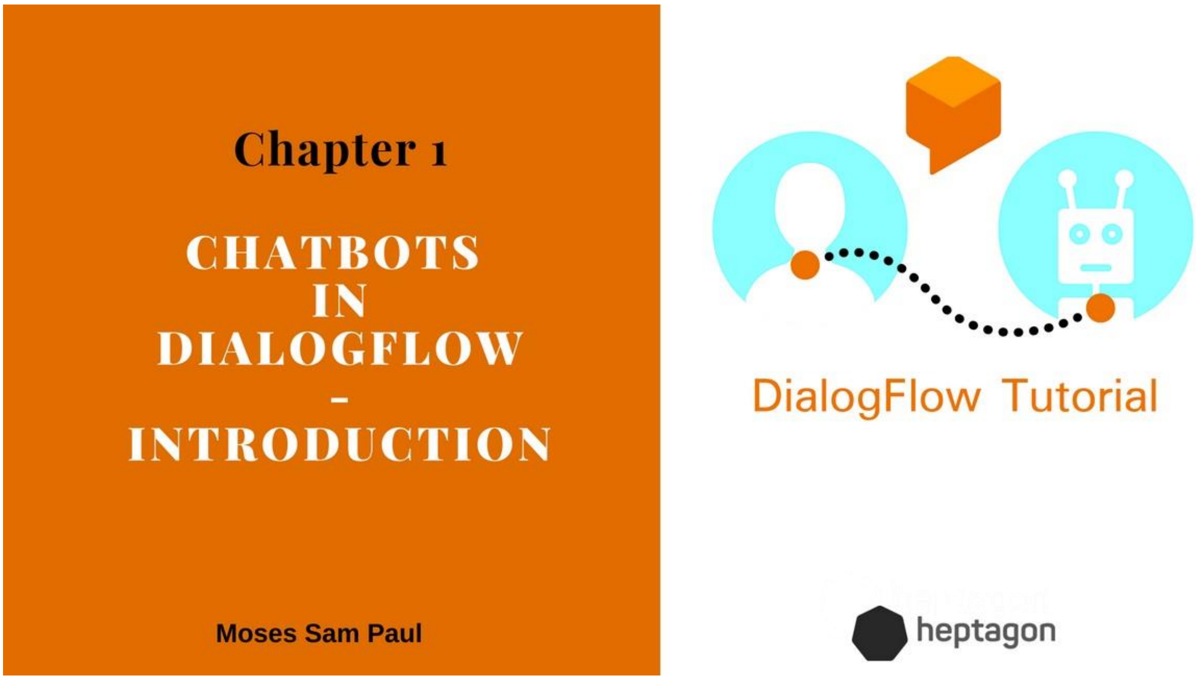
Chapter 1: Basic Gyaan about Chatbots and setting up the dialog flow environment

Chapter 2: Understanding the building blocks of Dialogflow

Chapter 3: Using linear, followup and non-linear dialogs to create a “TripPlanner” bot

Chapter 4: We created a new bot “StockTracker” bot to pull data from an external site through fulfilment and revert the result to the user

Chapter 5: Connect the “StockTracker” Chatbot with Slack



In this post, we’ll learn

- Why build a chatbot?
- Pre-requisites
- Introduction to Dialogflow
- Flow of conversation
- Setting up dialogflow account

- Authorise Dialogflow on Google Cloud

Why build a chatbot?

A chatbot is, in essence, a piece of robotic software used to imitate human conversation through text chats and voice commands (a good example being Siri or Amazon Alexa).

2 Types of chatbots:

1. Rule based chatbots (if you ask for phones the relevant phone pages open up in an e-commerce site that’s an example of a rule based chatbot)
2. A.I. based chat bots (learn over a period of time using Machine Learning techniques) —dialog flow is an example of that

Chatbots are extremely valuable for businesses and this value will only increase as time goes by.

On obvious area of chatbot implementation is customer service. Bots are invaluable here. Waiting on hold may soon be a thing of the past as they become advanced enough to deal with basic level customer service queries, and this is already being used by a lot of companies worldwide. Nordstrom, for example, implemented a chatbot to assist with customer service at the end of 2016, and this has made their technical support much more responsive and immediate. It’s no secret that this has resulted in significant cost reduction.

Text and Voice based chatbots are the future and if you are an entrepreneur or a techie, it’s the right time to spend some time learning about building these bots.

Pre-requisites

No programming experience is required as this series is mostly gonna be using GUI(Graphical User Interface) of Google’s dialogflow.

Although, in chapter 4, we will be using a little bit of programming to pull data from a stockmarket data site to display the result via our chatbot. If you have some experience in dabbling with **javascript /Restful APIs** it might help. Even if you don’t have any experience, don’t fret will break it down.

Introduction to Dialogflow

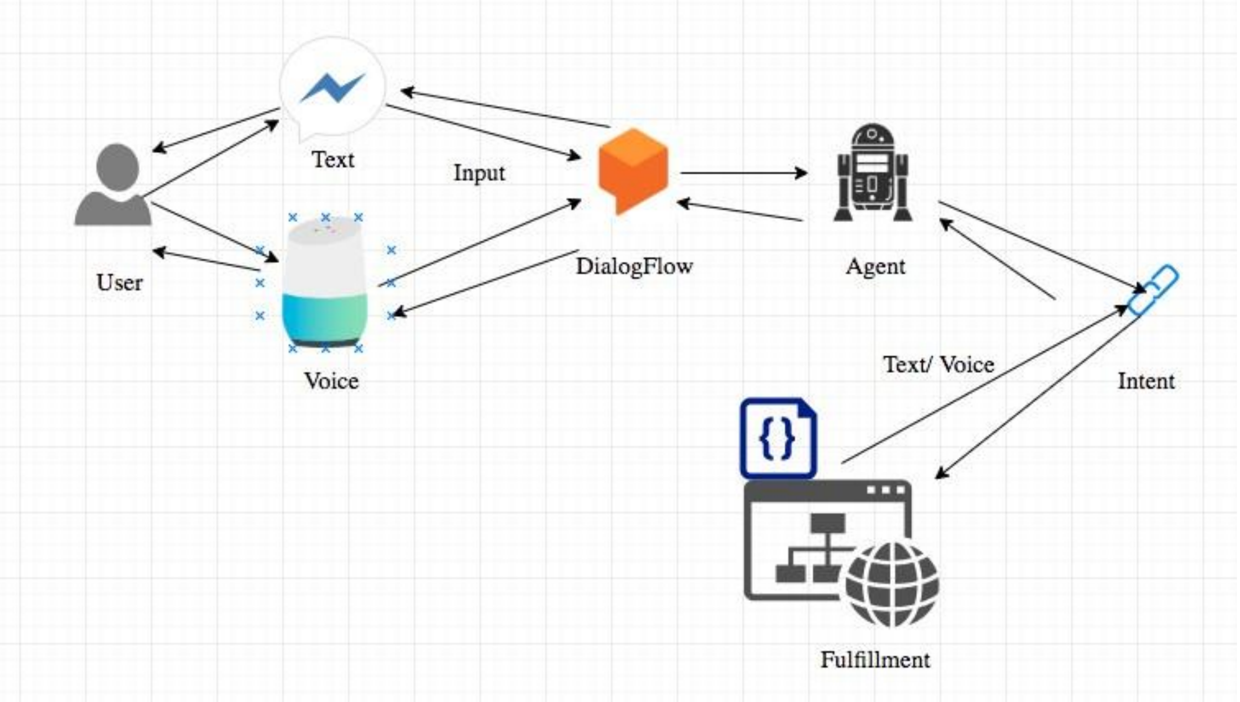
> **Dialogflow** (formerly Api.ai, Speaktoit) is a Google-owned developer of human–computer interaction technologies based on natural language conversations. The company is best known for creating the Assistant (by Speaktoit), a virtual buddy for Android, iOS, and Windows Phone smartphones that performs tasks and answers users’ question in a natural language. [1] Speaktoit has also created a natural language processing engine that incorporates conversation context like dialogue history, location and user preferences.—Wiki

Supports 14+ languages in 16+ programming languages.

SDKs to work with web apps, mobile app & wearables

Integrates with 14+ chat platforms such as slack, messenger, Alexa, Google Home etc.

Flow of conversation within DialogFlow



User: We, Machines!

Text / Voice : The user interacts with an app like facebook messenger / google home to start the interaction with the bot.

Dialogflow: Bot platform

Agent: A module within dialogflow which incorporates Natural Language Processing to understand what the user meant and to figure out what “action” has to be carried out. The agent transforms the user request into machine readable actionable data.

Intent: Support or the service that the user wants from the agent. Intent is configured by the developers. Intent determines the action by the code.

Fulfillment: This is the code. This part of the conversation lets you pass on the request from your bot to an external source and get response and pass it back to the user. This is achieved via Webhook. Setting up a webhook allows you to pass information from a matched intent into a web service and get a result from it.

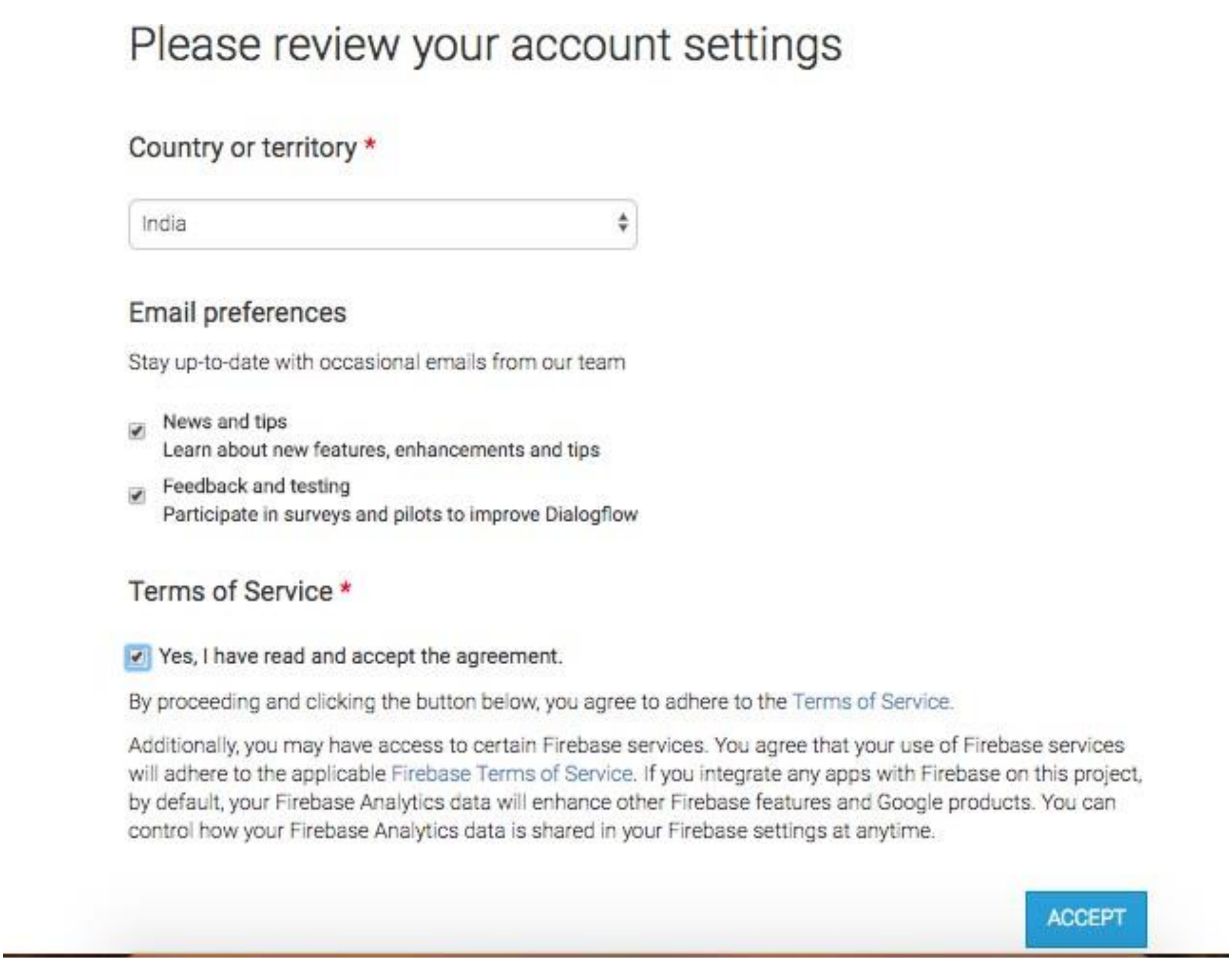
Note: Don’t be threatened by the terms here. Once we setup dialogflow account and open the site, all this will fall into place.

Setting up Dialogflow account.

Ok now that the boring ‘theory / lecture’ part is over, let’s jump in and start setting up the environment where we’ll be creating our bot!

1. Goto : https://dialogflow.com/

2. Create an account with a gmail account, and “agree” to the terms & conditions.



Please review your account settings

Country or territory *

India

Email preferences

Stay up-to-date with occasional emails from our team

☑ News and tips
Learn about new features, enhancements and tips

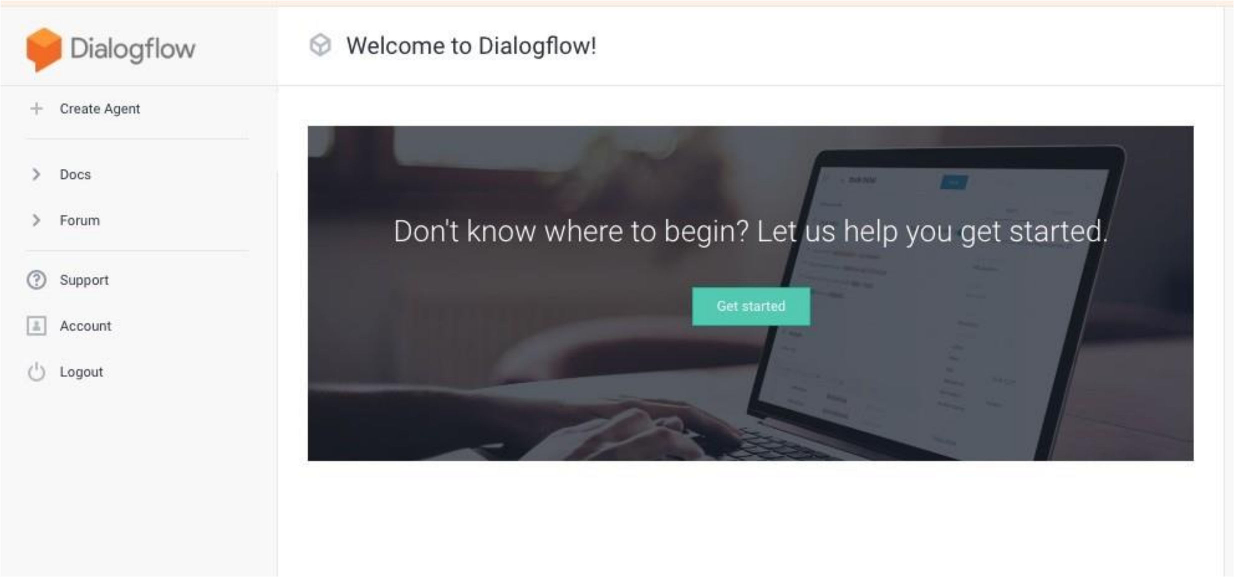
☑ Feedback and testing
Participate in surveys and pilots to improve Dialogflow

Terms of Service *

☑ Yes, I have read and accept the agreement.

By proceeding and clicking the button below, you agree to adhere to the Terms of Service.

Additionally, you may have access to certain Firebase services. You agree that your use of Firebase services will adhere to the applicable Firebase Terms of Service. If you integrate any apps with Firebase on this project, by default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your Firebase settings at anytime.



Welcome to Dialogflow!

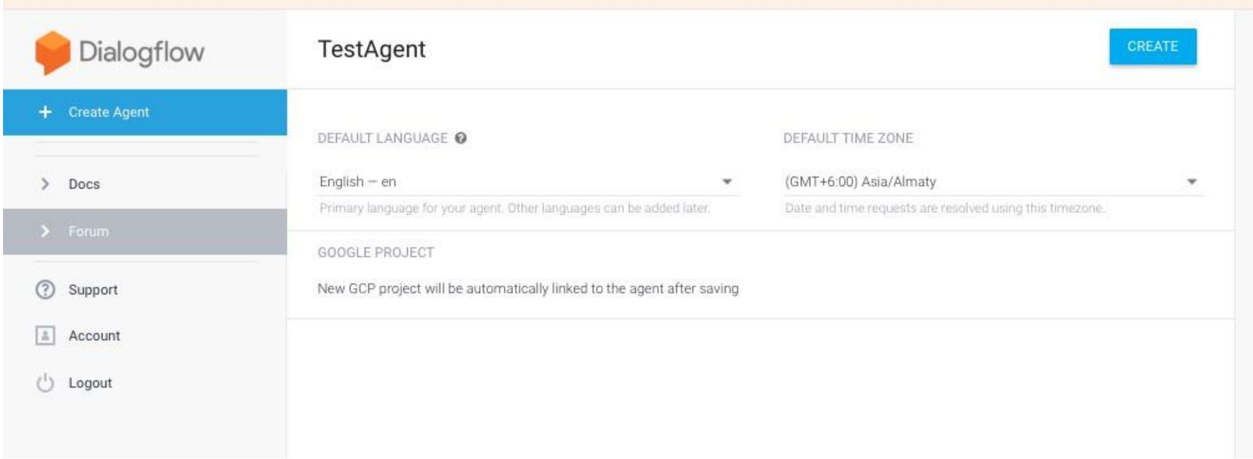
Don't know where to begin? Let us help you get started.

If you wanna delve deeper, do go through the docs here.

Overview | Dialogflow
*This tutorial shows you how to build a simple Dialogflow agent, walking you through the most important features of…*dialogflow.com

Authorise Dialogflow on Google Cloud

First step in creating our bot is to create an agent.



TestAgent

DEFAULT LANGUAGE: English — en
Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE: (GMT+6:00) Asia/Almaty
Date and time requests are resolved using this timezone.

GOOGLE PROJECT
New GCP project will be automatically linked to the agent after saving

This will create a new GoogleCloud Project automatically. If you are prompted for the authorisation, do allow. If you don't have a google cloud platform account please create one.

Summary:

- Chatbots built with Google's DialogFlow are intelligent personal assistants.
- Dialogflow abstracts out the Natural Language Processing, Machine Learning and other deeper concepts and gives a clean usable user interface to focus on the conversation flow and build bots.

In the next chapter, we'll understand the building blocks of dialogflow and start building our bot.

How to build a chatbot with Dialog flow | Chapter 2— Building Blocks of DialogFlow

In chapter 1, we covered the importance of chatbots and setting up the dialogflow account. In this chapter we will understand the building blocks of dialogflow.

Series Content:

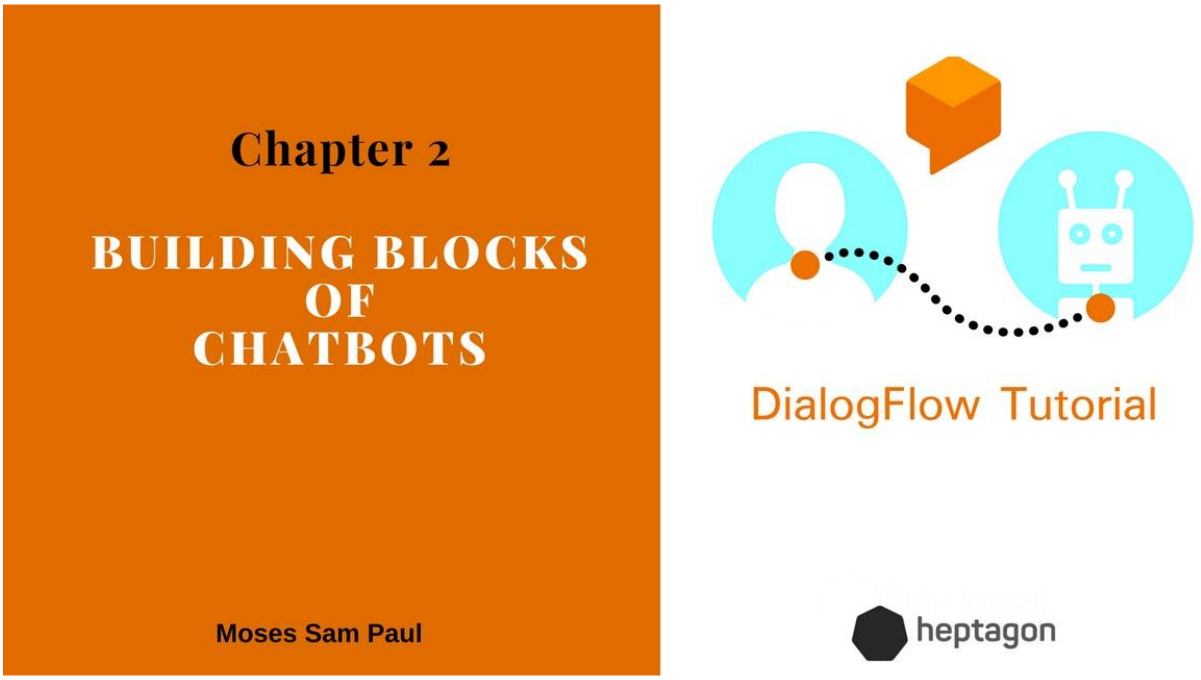
Chapter 1: Basic Gyaan about Chatbots and setting up the dialog flow environment

Chapter 2: Understanding the building blocks of Dialogflow

Chapter 3: Using linear, followup and non-linear dialogs to create a "TripPlanner" bot

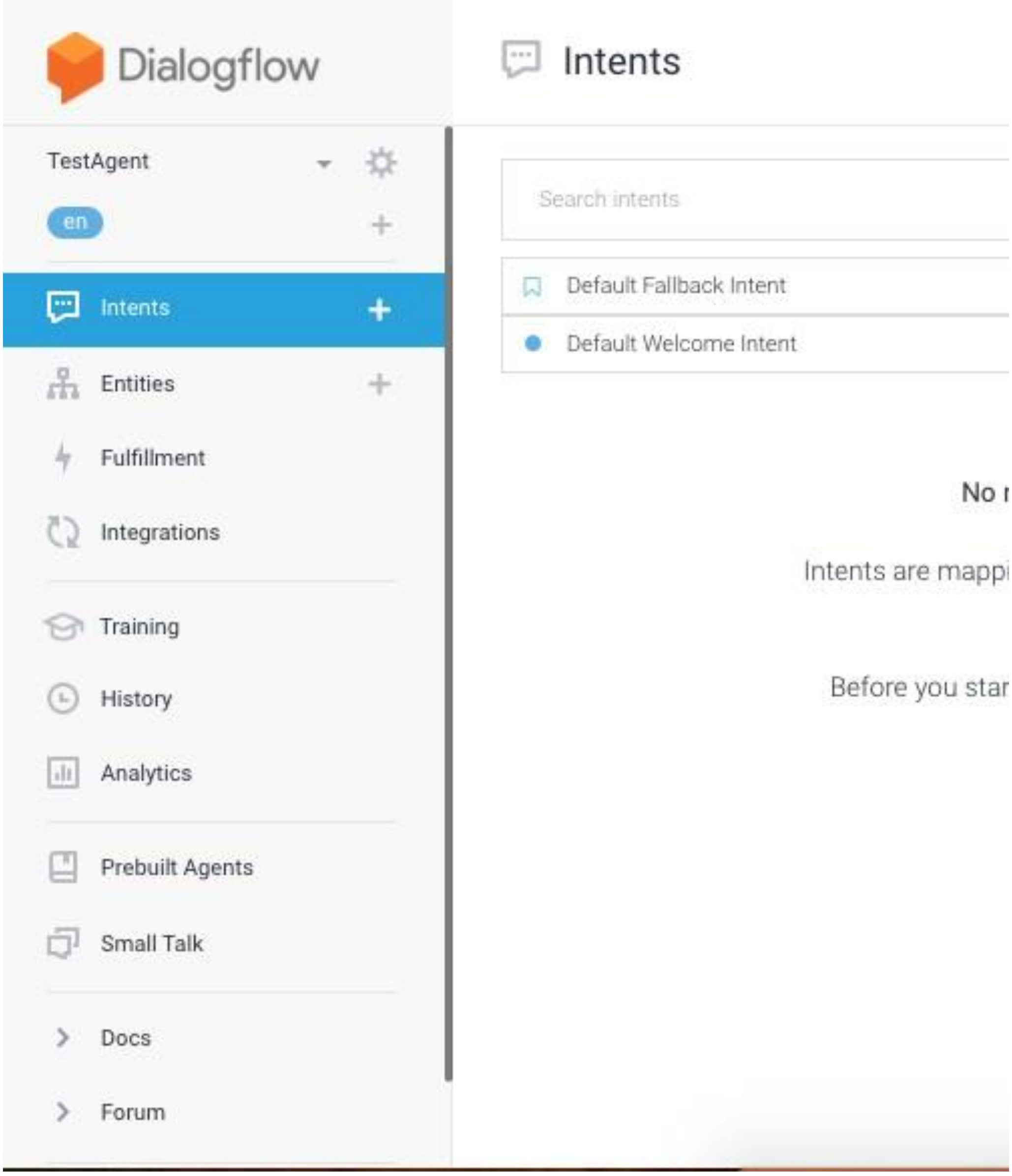
Chapter 4: We created a new bot "StockTracker" bot to pull data from an external site through fulfilment and revert the result to the user

Chapter 5: Connect the "StockTracker" Chatbot with Slack



Re-cap:

Building blocks of Dialogflow.



1. **Agents** : Help convert user request into actionable data. Eg: TestAgent in the image above
2. **Intents**: These are configured by developers which indicate what the objective of the user might be when he/she/she makes a specific request. Eg: Book a flight / Collect Feedback etc
3. **Entities**: Help extract information from user speech with the help of prompts. Eg: “Book a flight” intent might need such as the: to and from cities, date, class etc as entities that the agent tries to extract from the user via conversations. The information received here are sent on for fulfilment.
4. **Fulfilment**: Code that fulfils the intent of the user’s request.
5. **Integrations**: Twitter, Slack, Googlehome etc..
6. **Prebuilt Agents:** Check out the various built-in agents
7. **Smalltalk:** Helps make bots friendly and chatty with no coding from our end.

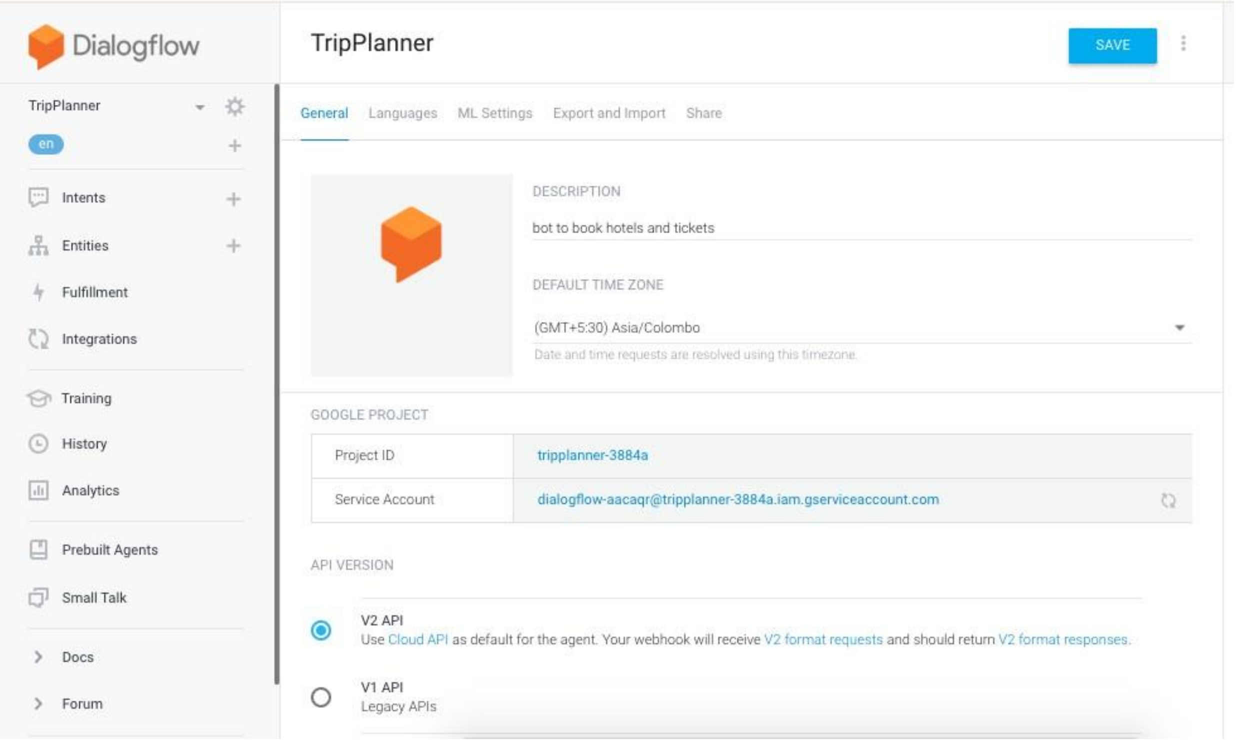
Agents

Agents translate user requests to actionable data i.e. intents. It’s essentially a module within dialog flow which incorporates Natural Language Processing to understand what the user meant and to figure out what “action” has to be carried out.

Agents manage conversations with the user through intent, entities, contexts and other building blocks.

```
We are gonna build a bot that helps us with planning a business trip
```

Build the Agent : Trip Planner



TripPlanner

DESCRIPTION
bot to book hotels and tickets

DEFAULT TIME ZONE
(GMT+5:30) Asia/Colombo
Date and time requests are resolved using this timezone.

GOOGLE PROJECT

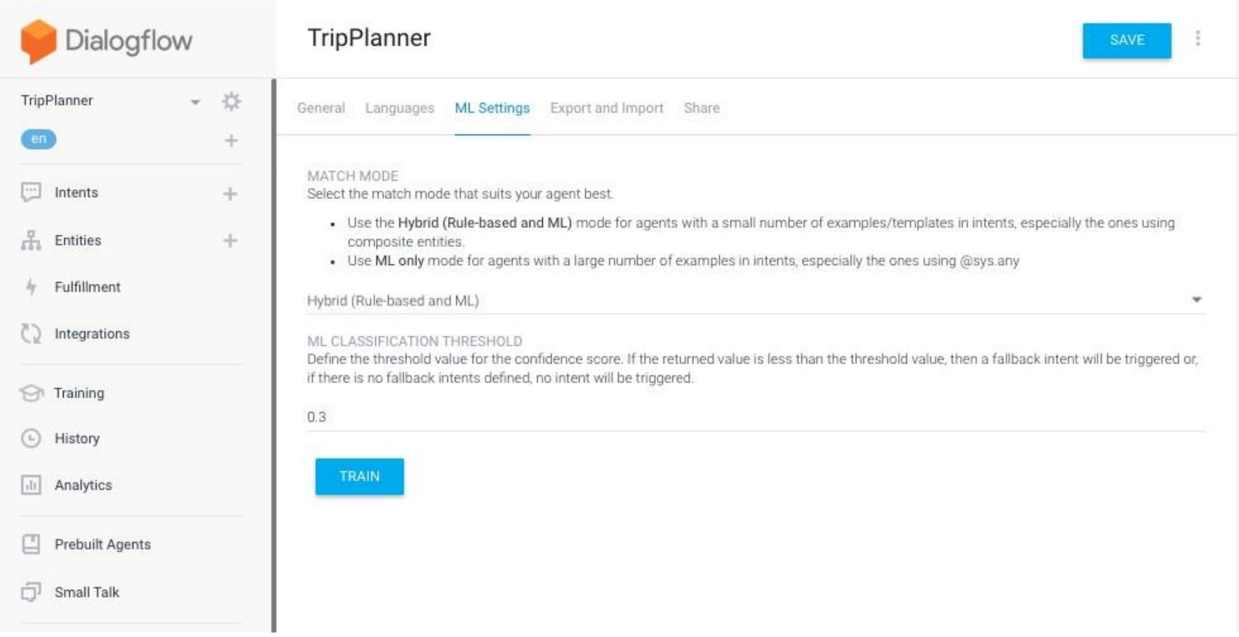
Project ID	tripplanner-3884a
Service Account	dialogflow-aacaqr@tripplanner-3884a.iam.gserviceaccount.com

API VERSION

● V2 API
Use Cloud API as default for the agent. Your webhook will receive V2 format requests and should return V2 format responses.

○ V1 API
Legacy APIs

Explore the settings page…the ML settings gives us the type of machine learning we wanna use for this bot.



TripPlanner

MATCH MODE
Select the match mode that suits your agent best.

- Use the **Hybrid (Rule-based and ML)** mode for agents with a small number of examples/templates in intents, especially the ones using composite entities.
- Use **ML only** mode for agents with a large number of examples in intents, especially the ones using @sys.any

Hybrid (Rule-based and ML)

ML CLASSIFICATION THRESHOLD
Define the threshold value for the confidence score. If the returned value is less than the threshold value, then a fallback intent will be triggered or, if there is no fallback intents defined, no intent will be triggered.

0.3

ML CLASSIFICATION THRESHOLD — 0.3

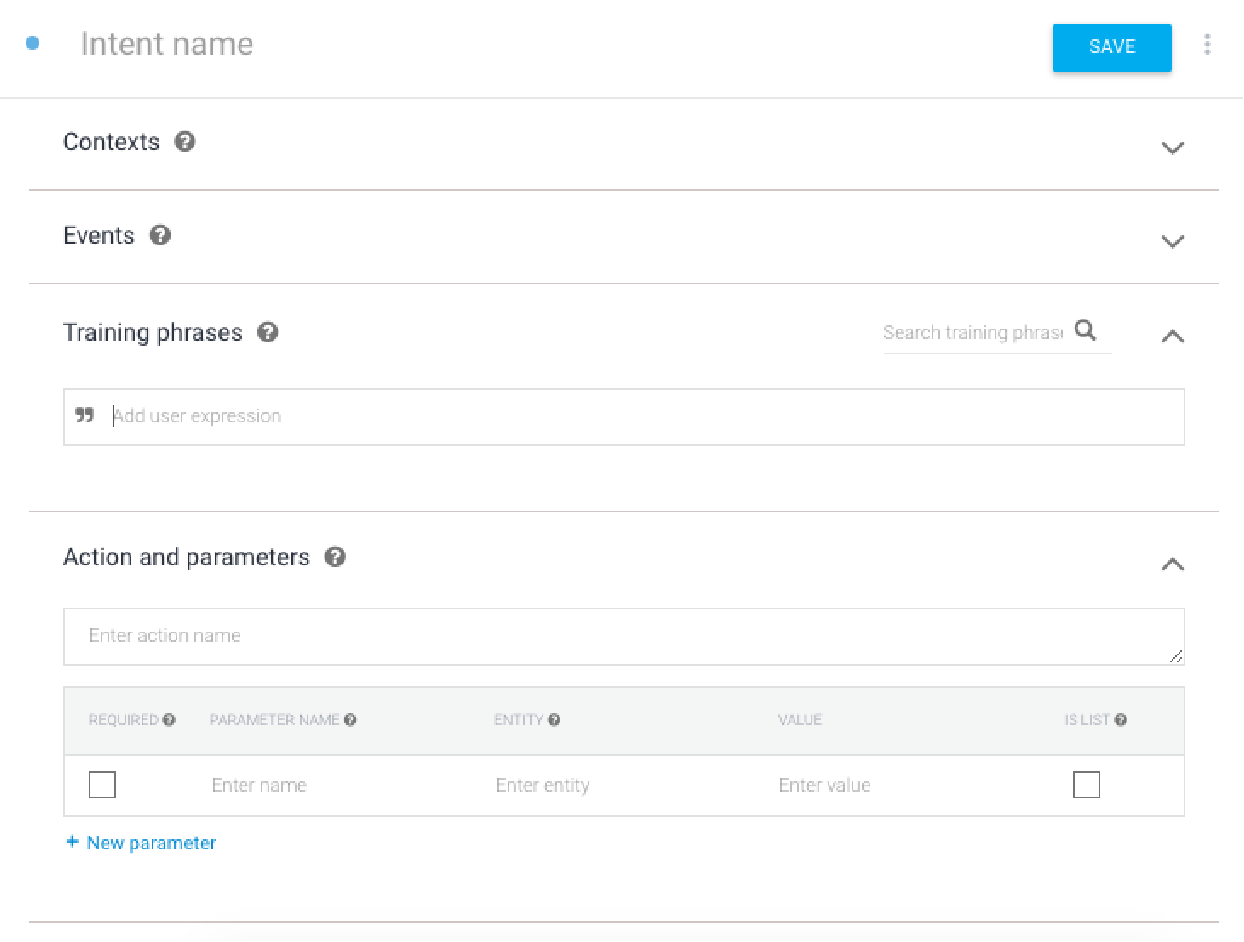
Define the threshold value for the confidence score. If the returned value is less than the threshold value, then a fallback intent will be triggered or, if there is no fallback intents defined, no intent will be triggered.

In simple terms, if the match is less than 30% percent then the inbuilt “fallback intent” (sorry, I don’t understand etc) will be triggered.

Try to go through the settings tab and the google cloud platform project that has been created automatically the moment you created the agent.

Intent

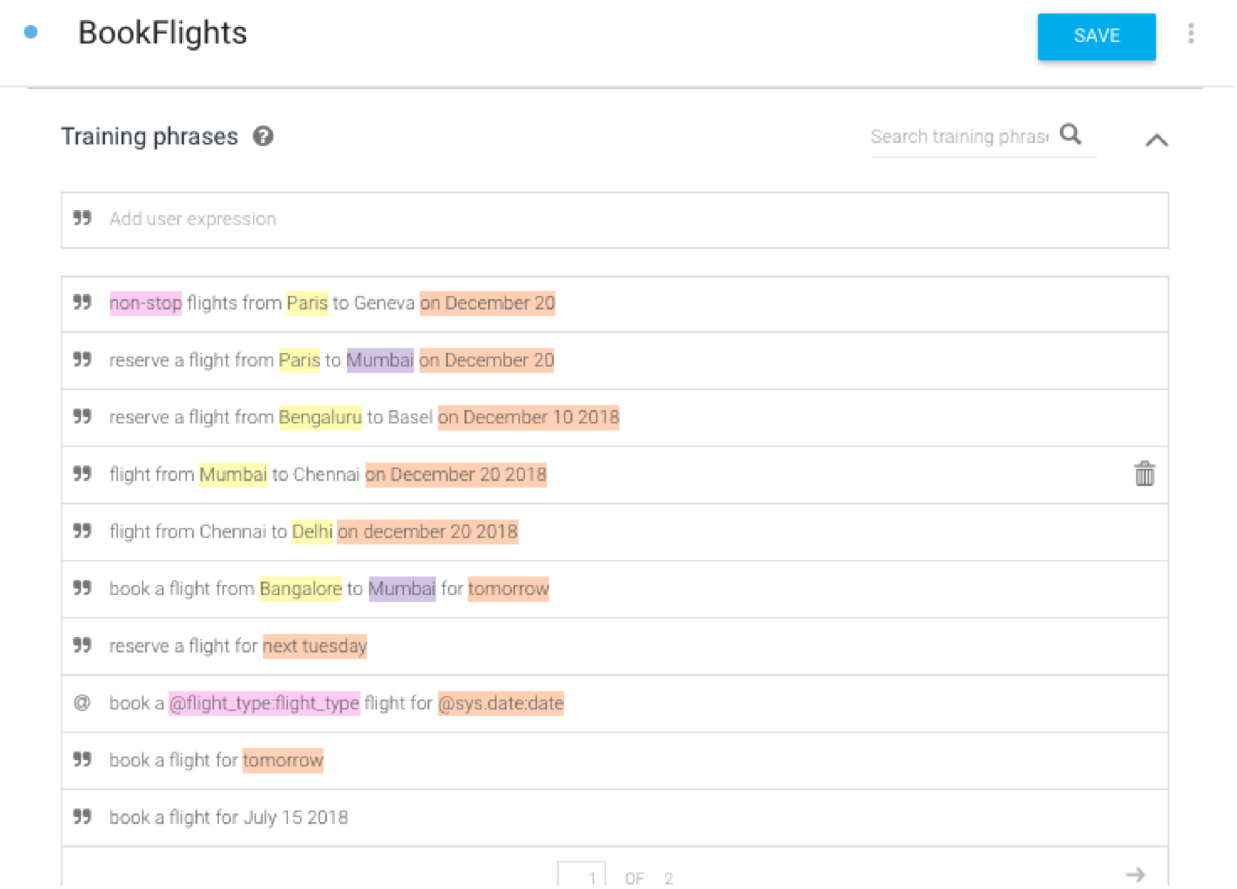
Intents are configured by developers and used to determine the action taken by the code. Think of using Intent as a mapping….what a user says and what your software should execute. <Don’t rush to create these intents…am just explaining stuff here, once we reach custom intent, will show how these intents are created>



Don’t worry about context and events for now. We’ll get there in chapter 3.

Training Phrase:

Phrases you can expect from the user that will trigger the intent.



BookFlights

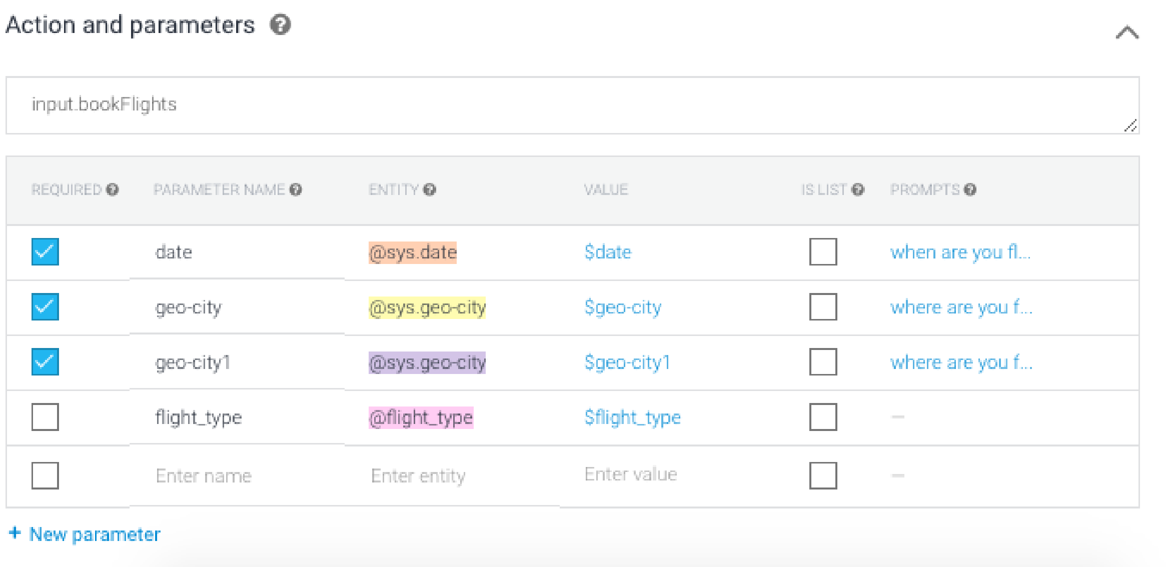
Training phrases

Add user expression

- non-stop flights from Paris to Geneva on December 20
- reserve a flight from Paris to Mumbai on December 20
- reserve a flight from Bengaluru to Basel on December 10 2018
- flight from Mumbai to Chennai on December 20 2018
- flight from Chennai to Delhi on december 20 2018
- book a flight from Bangalore to Mumbai for tomorrow
- reserve a flight for next tuesday
- book a @flight_type:flight_type flight for @sys.date:date
- book a flight for tomorrow
- book a flight for July 15 2018

Action & Parameters:

The inputs you might need from the user to take an action on the user request. For eg: In BookFlights intent, we might need the To city, From City, Date etc to finish the action. We have to set entities first for it to be mapped here but go with the flow for now.

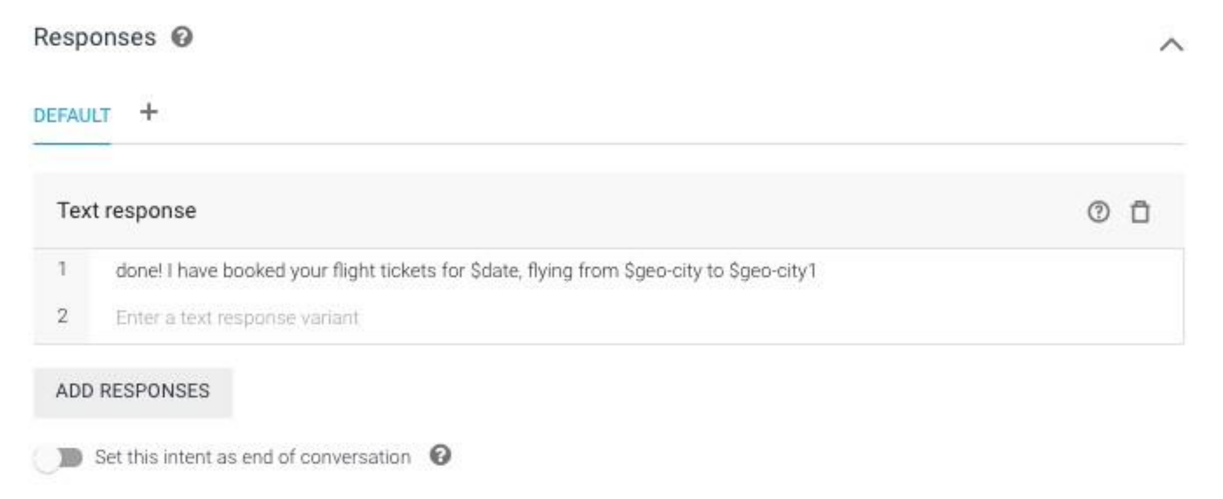


Action and parameters

input.bookFlights

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	date	@sys.date	$date	□	when are you fl...
☑	geo-city	@sys.geo-city	$geo-city	□	where are you f...
☑	geo-city1	@sys.geo-city	$geo-city1	□	where are you f...
□	flight_type	@flight_type	$flight_type	□	–
□	Enter name	Enter entity	Enter value	□	–

Response:



1 done! I have booked your flight tickets for $date, flying from $geo-city to $geo-city1

2 Enter a text response variant

Set this intent as end of conversation

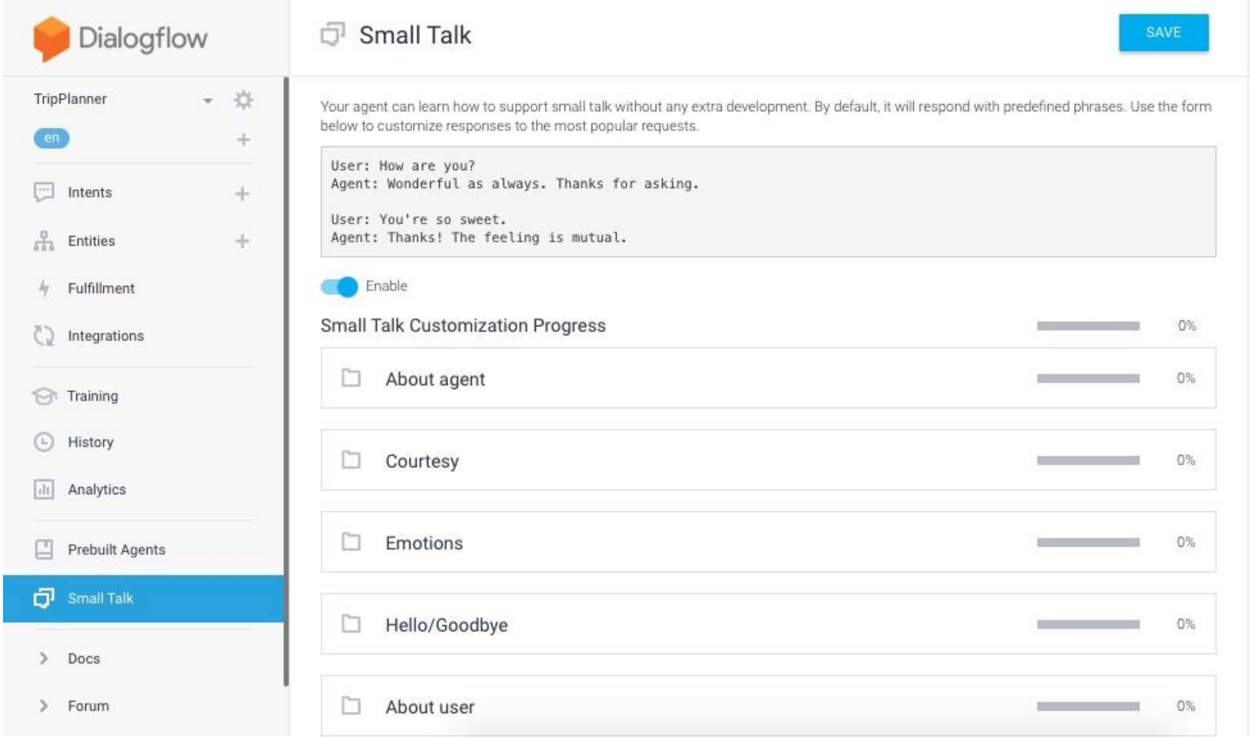
Default Intents:

Dialogflow has few default intents that can help us save a lot of time.

1. **Welcome Intent**: Greets the usr, exchange pleasantries (partially configured)
2. **Fallback Intent** : Default fall through intent when no others match (excuse me, I din’t get what you said)

Small Talk Intent:

Helps make bots friendly and chatty with no coding from our end. Again, this comes by default you just have to enable it.



Your agent can learn how to support small talk without any extra development. By default, it will respond with predefined phrases. Use the form below to customize responses to the most popular requests.

```
User: How are you?
Agent: Wonderful as always. Thanks for asking.

User: You're so sweet.
Agent: Thanks! The feeling is mutual.
```

Enable

Small Talk Customization Progress 0%

About agent 0%

Courtesy 0%

Emotions 0%

Hello/Goodbye 0%

About user 0%

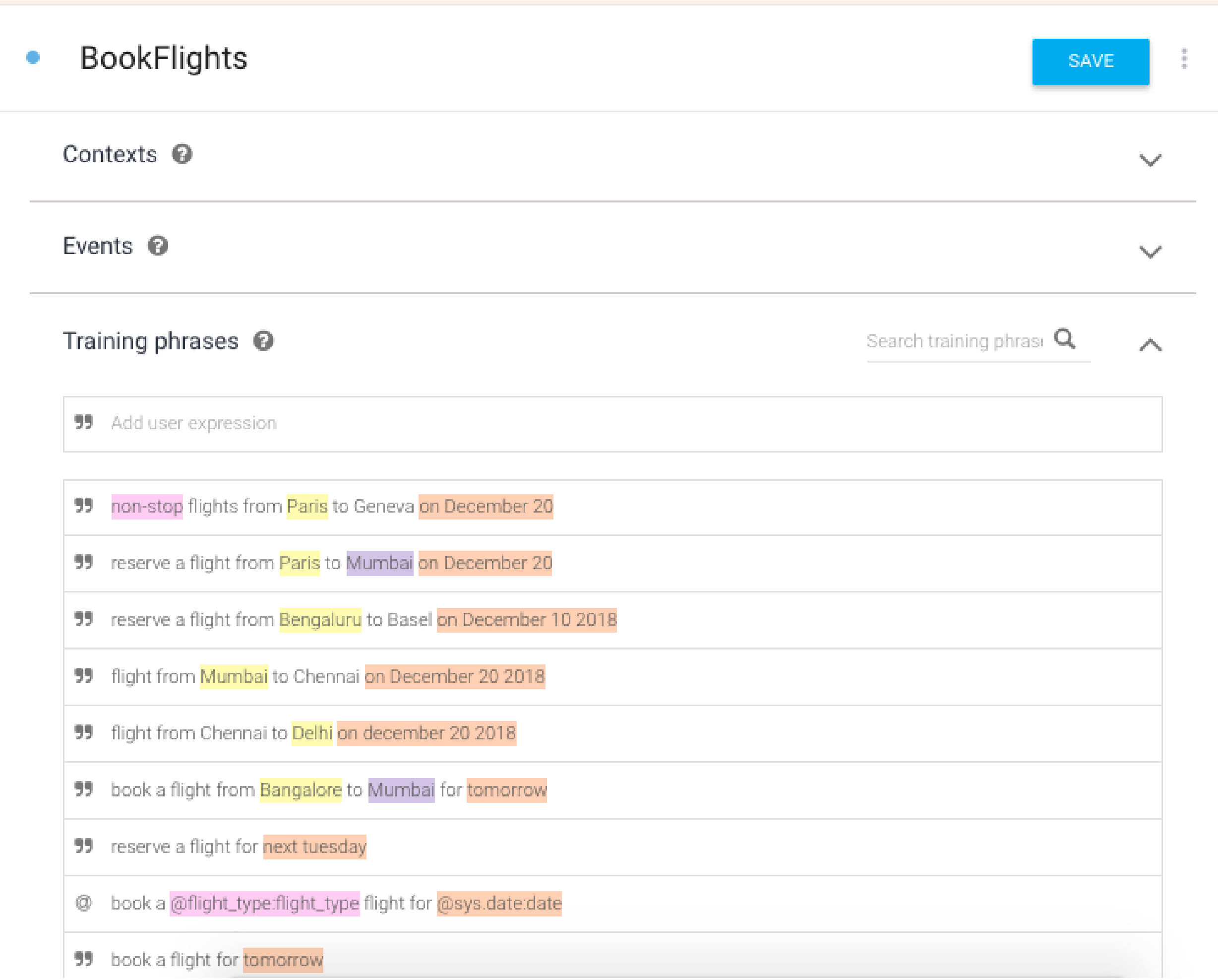
Custom Intent:

```
Continuing our effort to create the TripPlanner bot...we gotta create the intents
for the same. We are gonna help users book a flight, book a room and book a car so
we need to create 3 custom intents
```

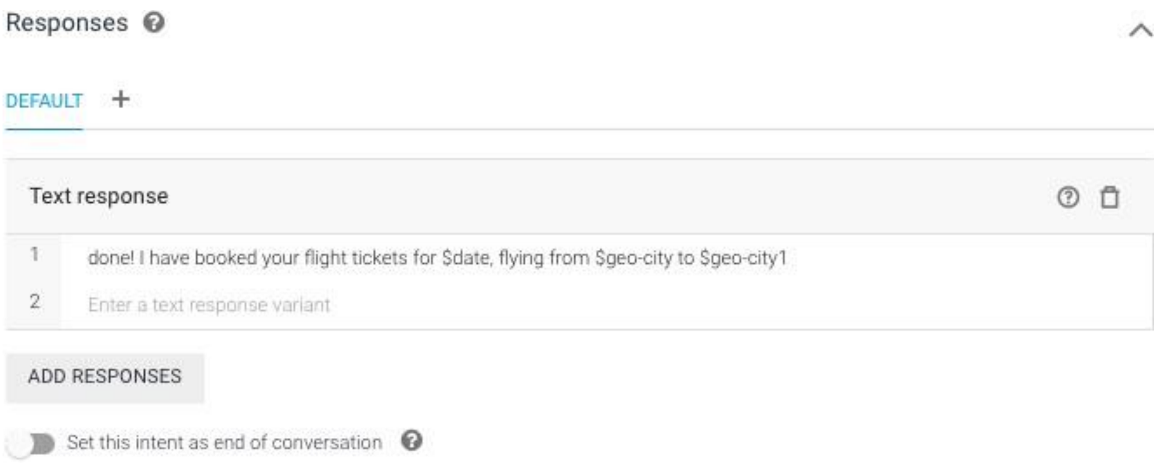
create custom intents

So we gotta create 3 custom intents.

1. BookFlights



- non-stop flights from Paris to Geneva on December 20
- reserve a flight from Paris to Mumbai on December 20
- reserve a flight from Bengaluru to Basel on December 10 2018
- flight from Mumbai to Chennai on December 20 2018
- flight from Chennai to Delhi on december 20 2018
- book a flight from Bangalore to Mumbai for tomorrow
- reserve a flight for next tuesday
- book a @flight_type:flight_type flight for @sys.date:date
- book a flight for tomorrow



Responses

DEFAULT +

Text response

1 done! I have booked your flight tickets for $date, flying from $geo-city to $geo-city1

2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

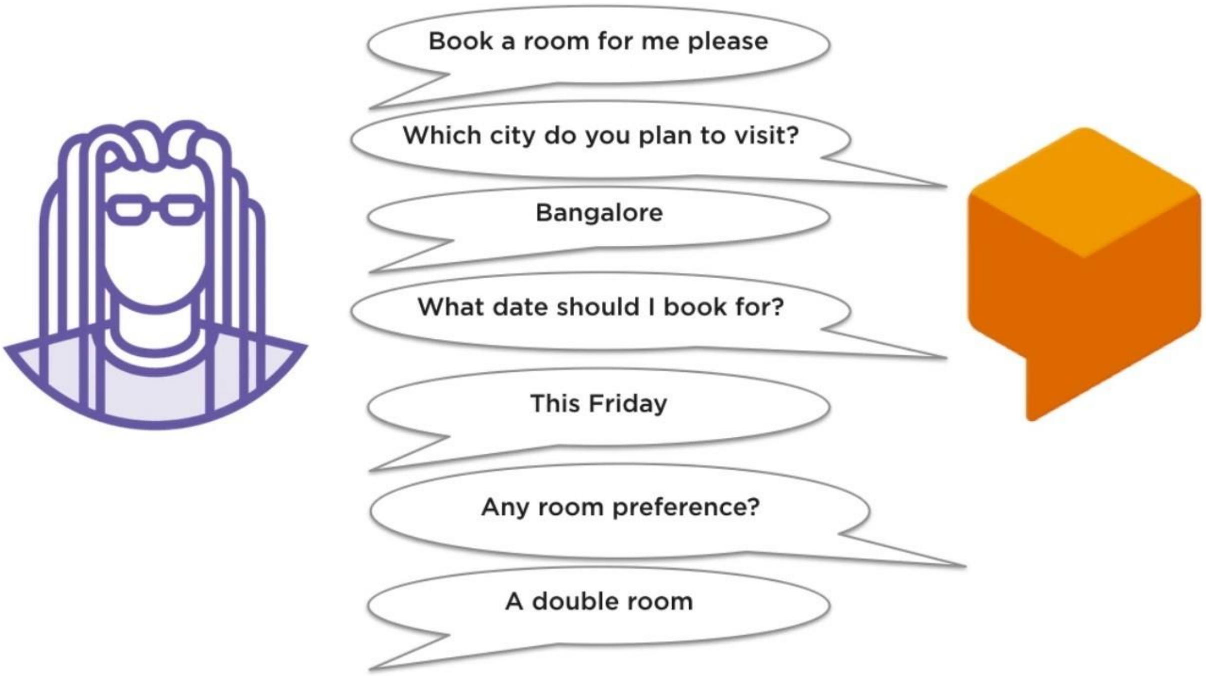
if the cities and the date aren't getting tagged / colored don't panic..it'll happen after we create the entities.

Similarly, create two more custom intents …. BookCars, BookRooms.

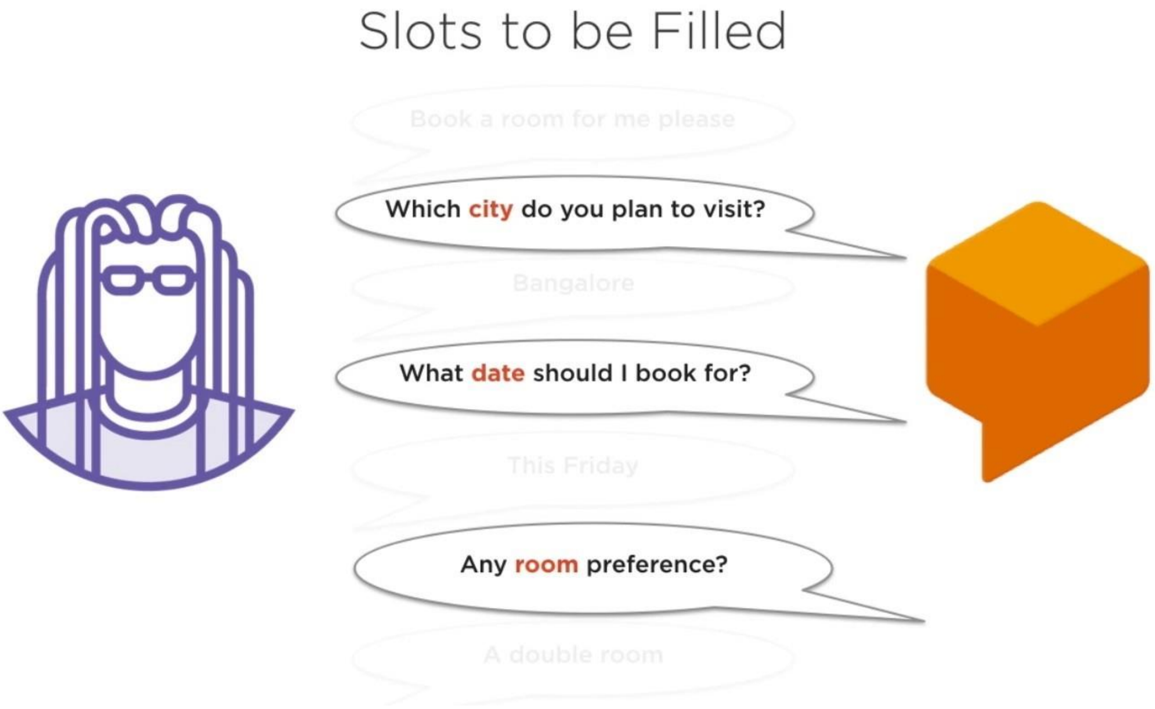
Entities

So, till now we have created one agent and three custom intents. Let's understand entities in this section.

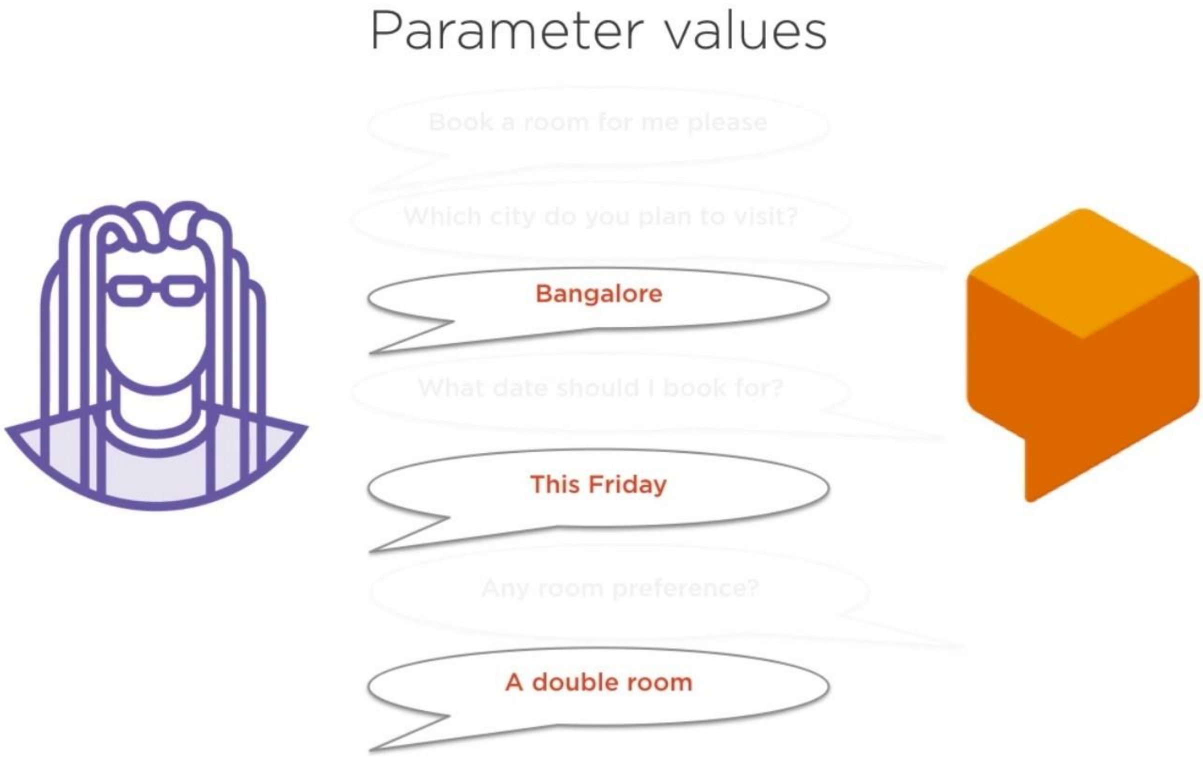
Entities are used to extract parameter values from user queries. So when the user says "Book me a flight" we usually ask for the city, date and the type of flight probably..these are called entities.


Book a room for me please
Which city do you plan to visit?
Bangalore
What date should I book for?
This Friday
Any room preference?
A double room

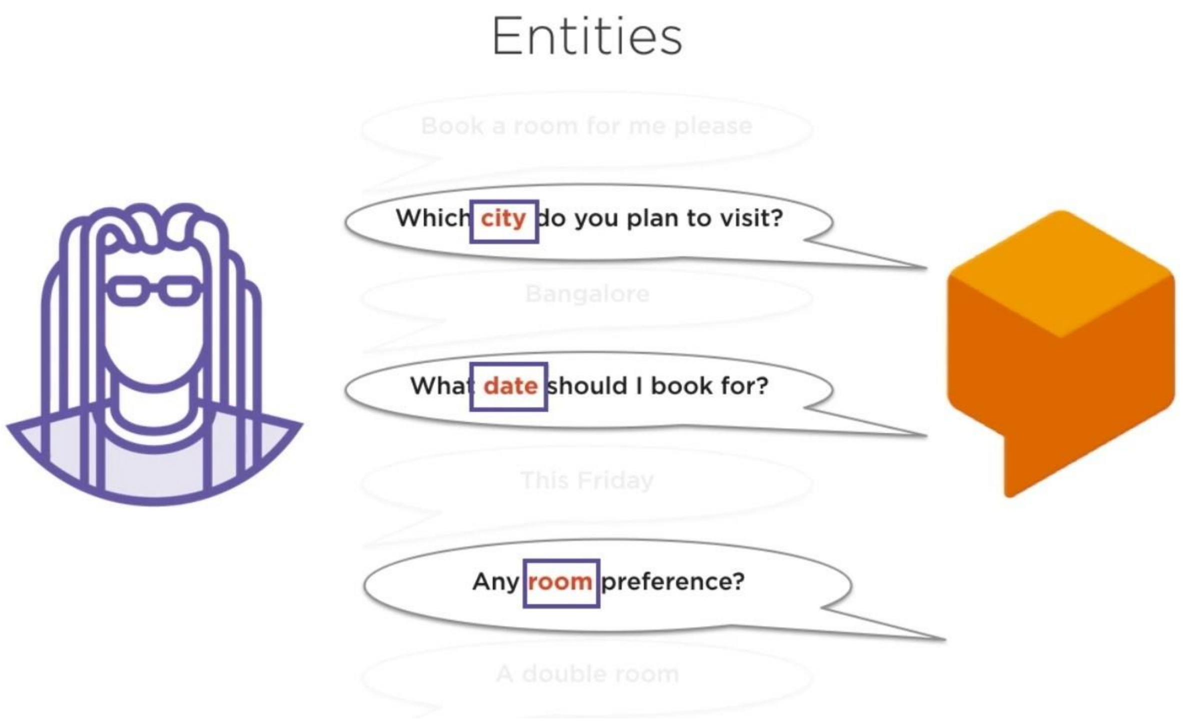
Example Conversation


Slots to be Filled
Book a room for me please
Which city do you plan to visit?
Bangalore
What date should I book for?
This Friday
Any room preference?
A double room

We (chatbot) asks questions to fill our Slots


Parameter values
Book a room for me please
Which city do you plan to visit?
Bangalore
What date should I book for?
This Friday
Any room preference?
A double room

Whatever the user responds to the slot based questions are the parameters


Entities
Book a room for me please
Which city do you plan to visit?
Bangalore
What date should I book for?
This Friday
Any room preference?
A double room

The generic abstract form for the parameters are the entities

Three types of entities:

1. System—Time, City, Date
2. Developer—Room type
3. User—defined at the session level (user’s playlist)..think of them as cookies in websites

---

Ok, time to work:

Create Developer Entities:

Book Flights Intent:

1. From City
2. To City
3. Date
4. Flight Type [optional]

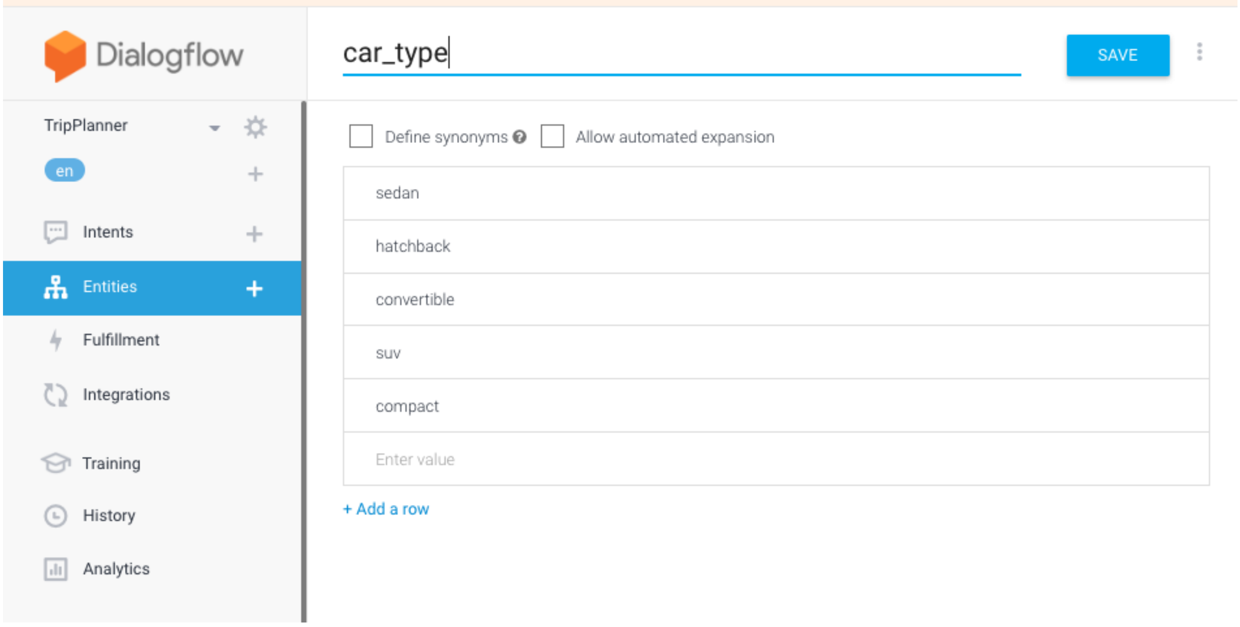
Book Rooms Intent:

1. City
2. Date
3. Room Type [Optional]

Book Cars Intent:

1. City
2. Date
3. Car Type

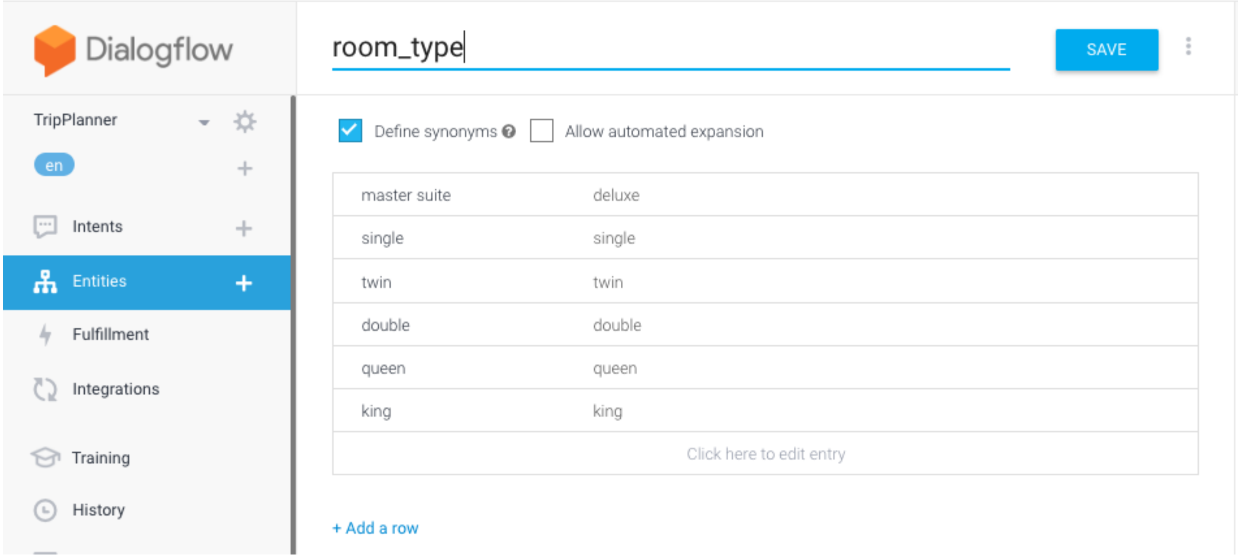
So as you can see City and Date are system entities so we don’t have to define them..but flight_type, room_type, car_type are all custom entities and we gotta define them..see images below.



car_type

□ Define synonyms □ Allow automated expansion

sedan
hatchback
convertible
suv
compact
Enter value

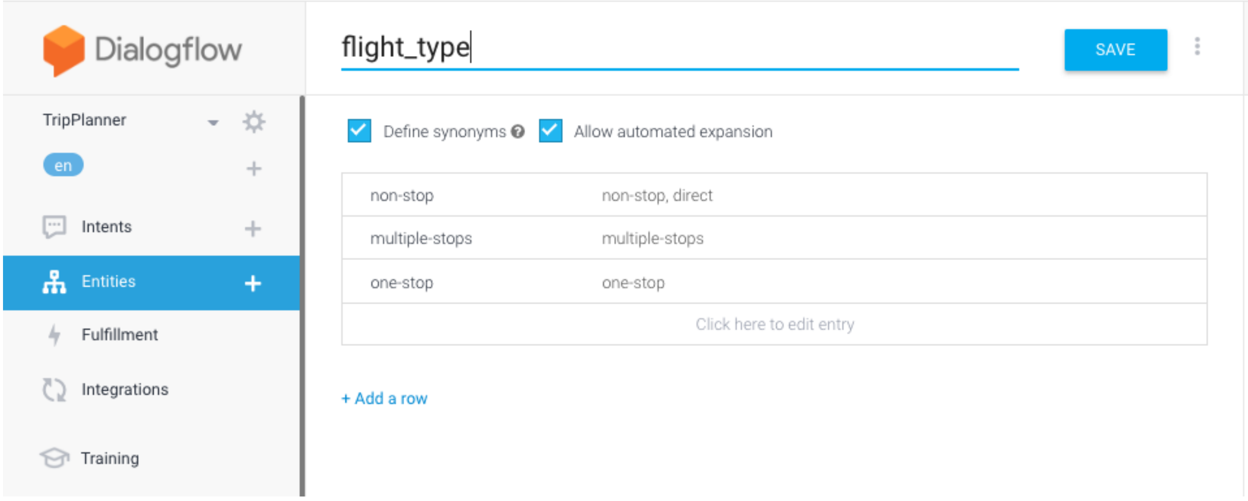


room_type

☑ Define synonyms □ Allow automated expansion

master suite	deluxe
single	single
twin	twin
double	double
queen	queen
king	king

custom entities are configured



flight_type

☑ Define synonyms ☑ Allow automated expansion

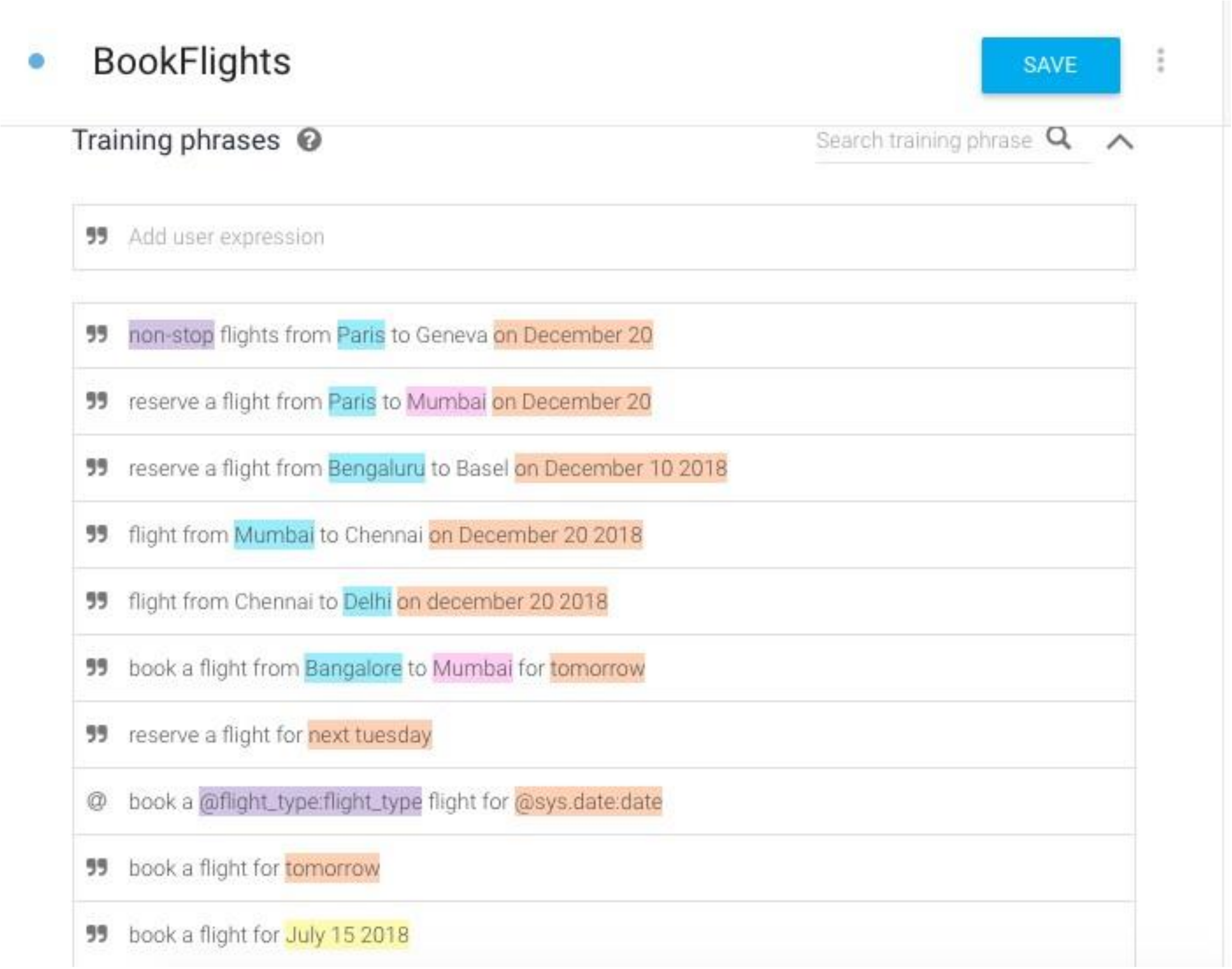
non-stop	non-stop, direct
multiple-stops	multiple-stops
one-stop	one-stop

Now that the Entities are defined lets go back and configure the intents we had created earlier.

Configuring Custom Intent

Book Flight intent with

- Expression with annotations
- parameters
- prompts



non-stop flights from Paris to Geneva on December 20

reserve a flight from Paris to Mumbai on December 20

reserve a flight from Bengaluru to Basel on December 10 2018

flight from Mumbai to Chennai on December 20 2018

flight from Chennai to Delhi on december 20 2018

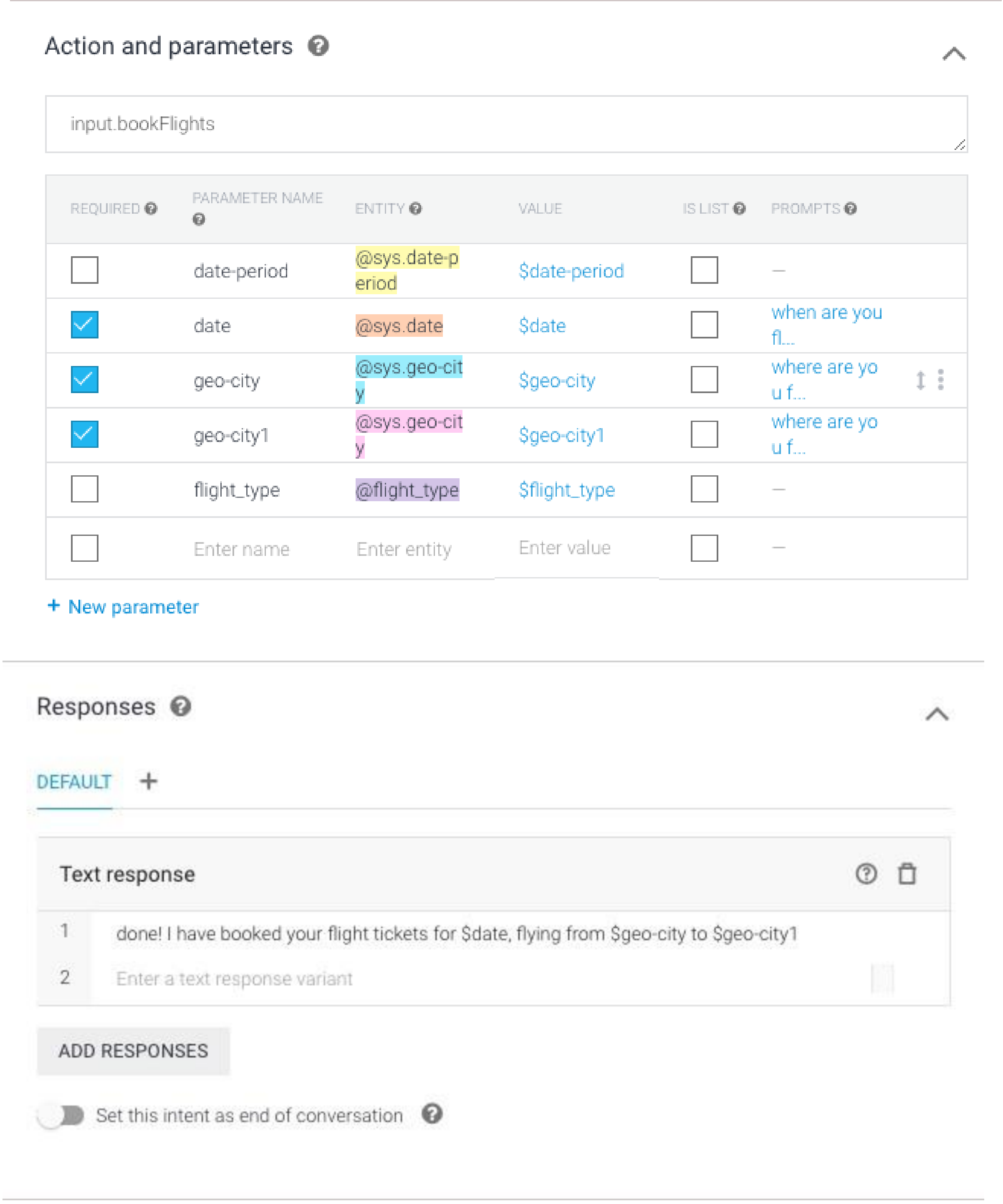
book a flight from Bangalore to Mumbai for tomorrow

reserve a flight for next tuesday

@ book a @flight_type:flight_type flight for @sys.date:date

book a flight for tomorrow

book a flight for July 15 2018



Action and parameters

input.bookFlights

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
□	date-period	@sys.date-period	$date-period	□	—
☑	date	@sys.date	$date	□	when are you fl...
☑	geo-city	@sys.geo-city	$geo-city	□	where are you f...
☑	geo-city1	@sys.geo-city	$geo-city1	□	where are you f...
□	flight_type	@flight_type	$flight_type	□	—
□	Enter name	Enter entity	Enter value	□	—

+ New parameter

Responses

DEFAULT +

Text response

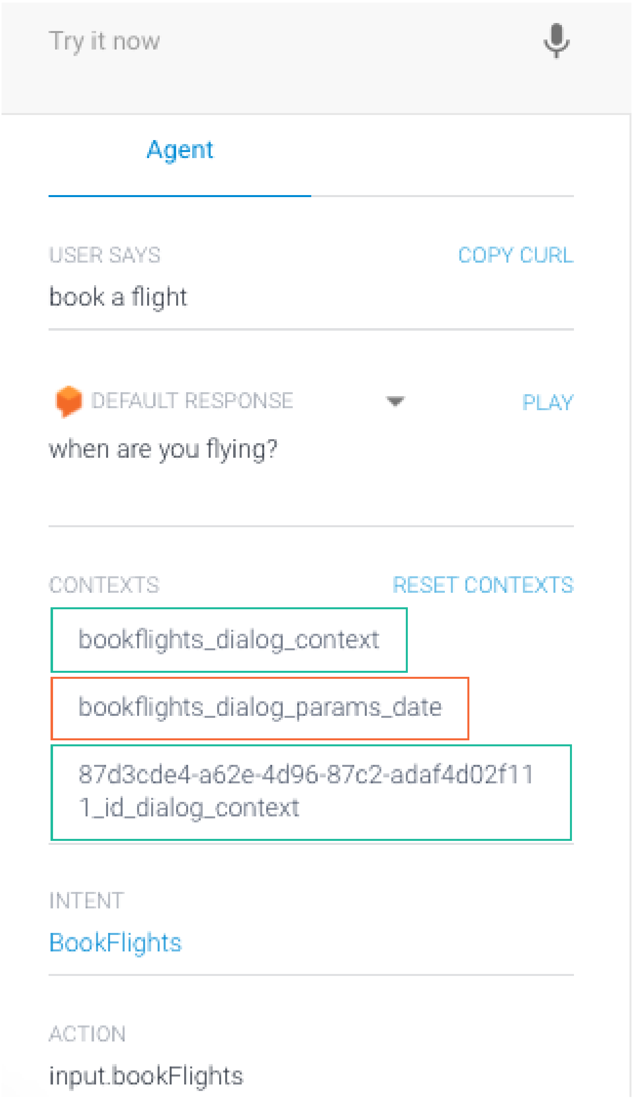
1 done! I have booked your flight tickets for $date, flying from $geo-city to $geo-city1

2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

Similarly, do the same for bookCars, bookRooms intents.



Try it now

Agent

USER SAYS

COPY CURL

book a flight

DEFAULT RESPONSE

PLAY

when are you flying?

CONTEXTS

RESET CONTEXTS

bookflights_dialog_context

bookflights_dialog_params_date

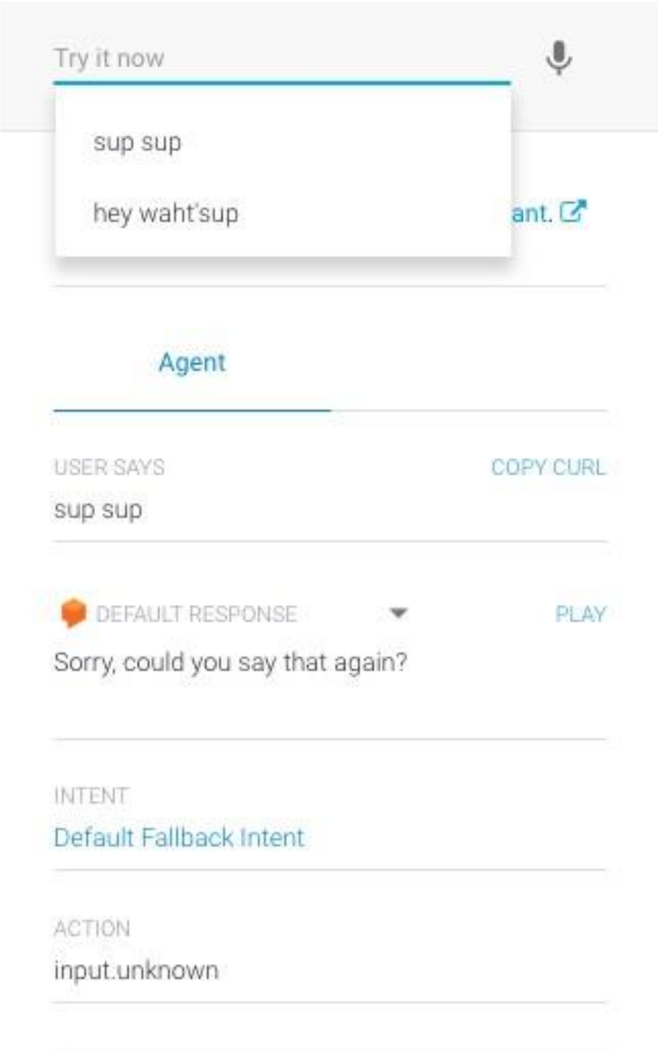
87d3cde4-a62e-4d96-87c2-adaf4d02f111_id_dialog_context

INTENT

BookFlights

ACTION

input.bookFlights



Try it now

sup sup

hey waht'sup

USER SAYS

sup sup

DEFAULT RESPONSE

Sorry, could you say that again?

INTENT

Default Fallback Intent

ACTION

input.unknown

Great, so what all have we done now?

- Created an Agent (TripPlanner)
- Created and Configured three Intents (BookFlights, BookCars, BookRooms)
- Created three custom entities. (flight_type, room_type, car_type)

Play around with the bot we have created till now, let me know if you face any difficulties.

In the next chapter, we shall delve a little deeper to understand the dialogs and how to maintain contexts between conversations.

How to build a chatbot with Dialog flow | Chapter 3— Dialogs for Conversations

In the last chapter we created a basic bot that can help us book flights, rooms and cars but did you notice that each time we had to give the city name and the date ..isn't that a little annoying ? Right.

So in this chapter we will look at dialogs and how it can help us make the bot a little smarter by retaining relevant information from the previous conversations with the users.

Series Content:

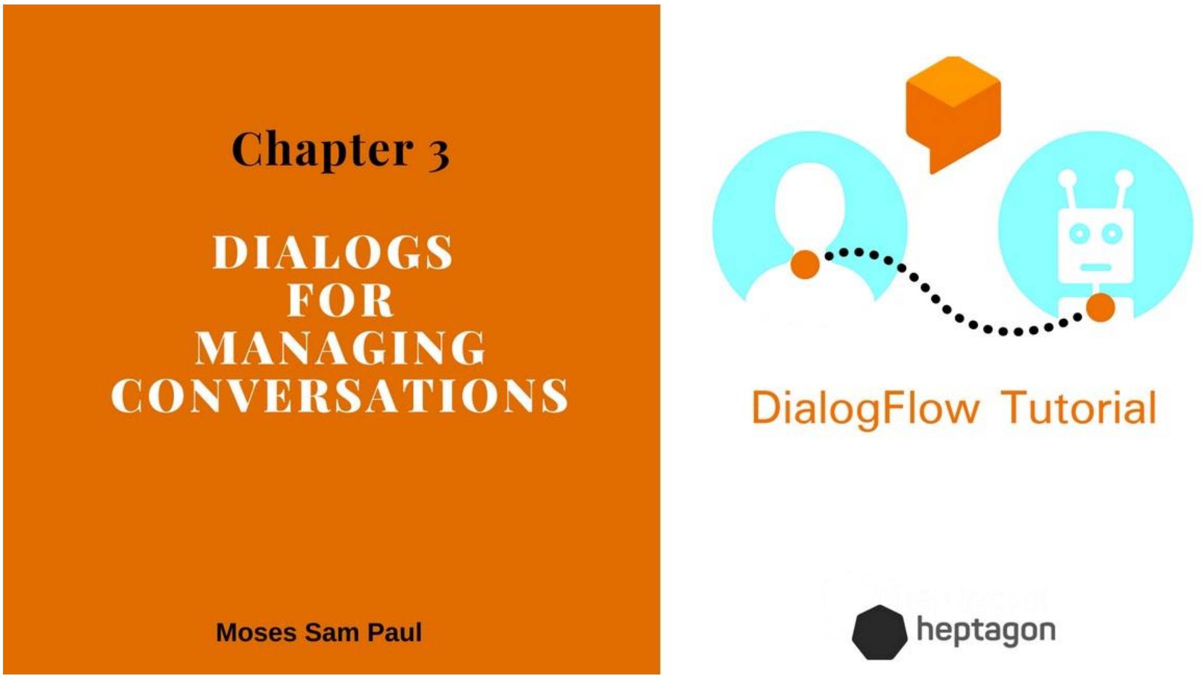
Chapter 1: Basic Gyaan about Chatbots and setting up the dialog flow environment

Chapter 2: Understanding the building blocks of Dialogflow

Chapter 3: Using linear, followup and non-linear dialogs to create a "TripPlanner" bot

Chapter 4: We created a new bot "StockTracker" bot to pull data from an external site through fulfilment and revert the result to the user

Chapter 5: Connect the "StockTracker" Chatbot with Slack

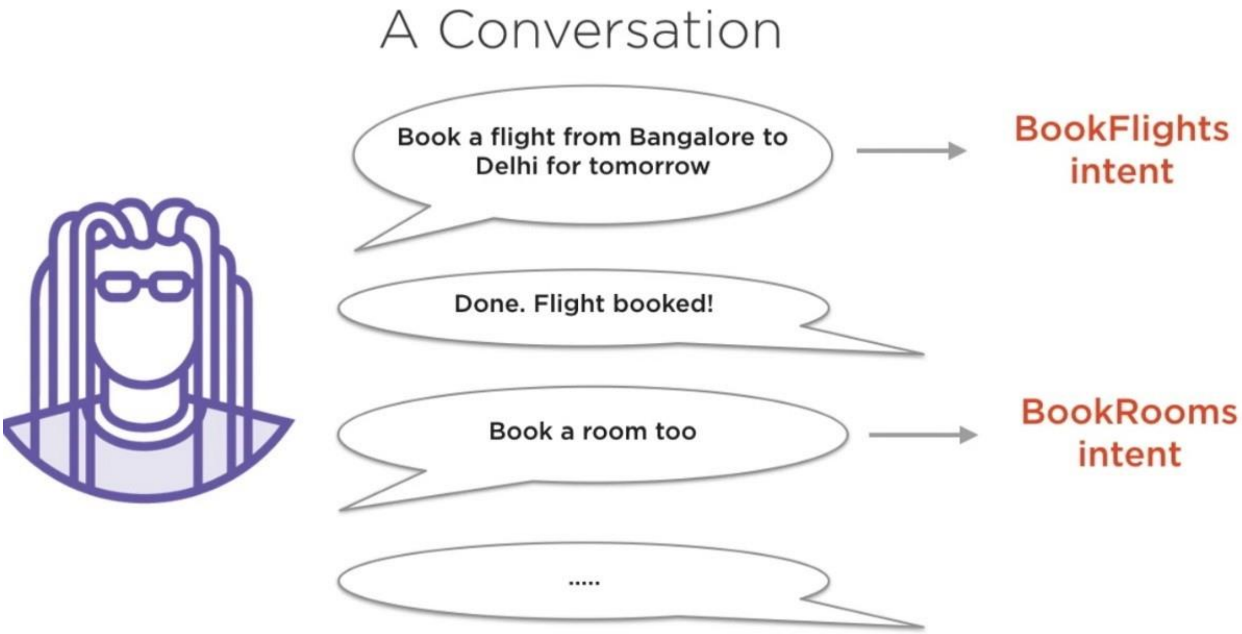


Using Dialogs for Managing Conversations

- Linear Dialogs can span a single intent or multiple intents
- Flow Predictably to elicit information needed to complete actions
- Contexts allow information sharing leading to more natural conversations
- Non-linear dialogs help branch to intents based on user responses

…book a room—when ? where? type ? these are linear dialogs to collect information from the user to fulfill

Context: lets us use linear dialogs that can be used across multiple intents



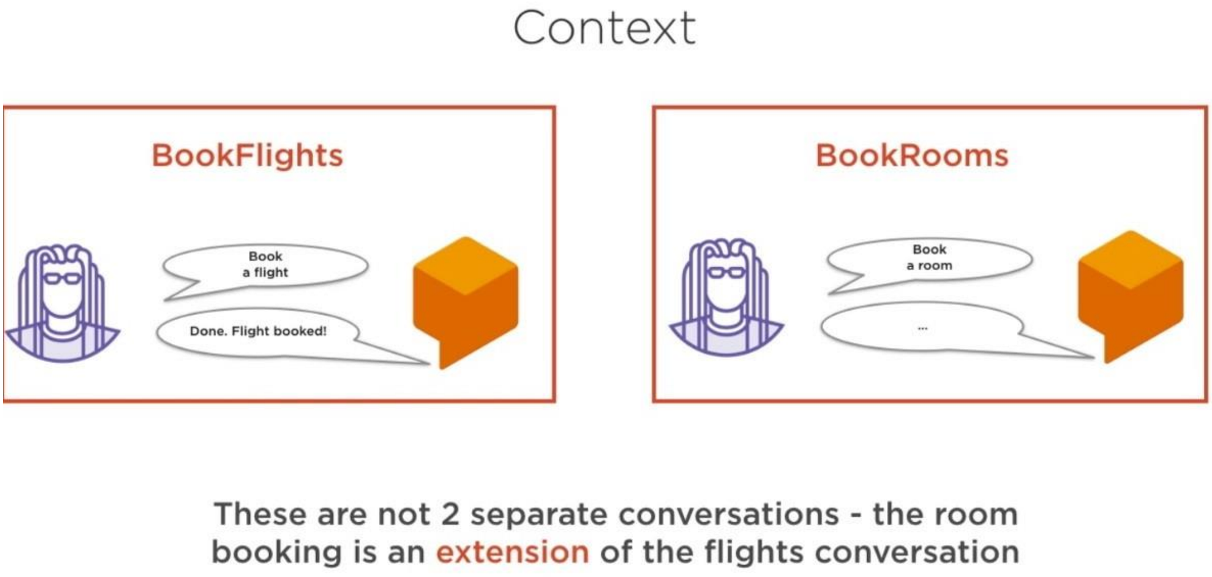
A Conversation

Book a flight from Bangalore to Delhi for tomorrow → BookFlights intent

Done. Flight booked!

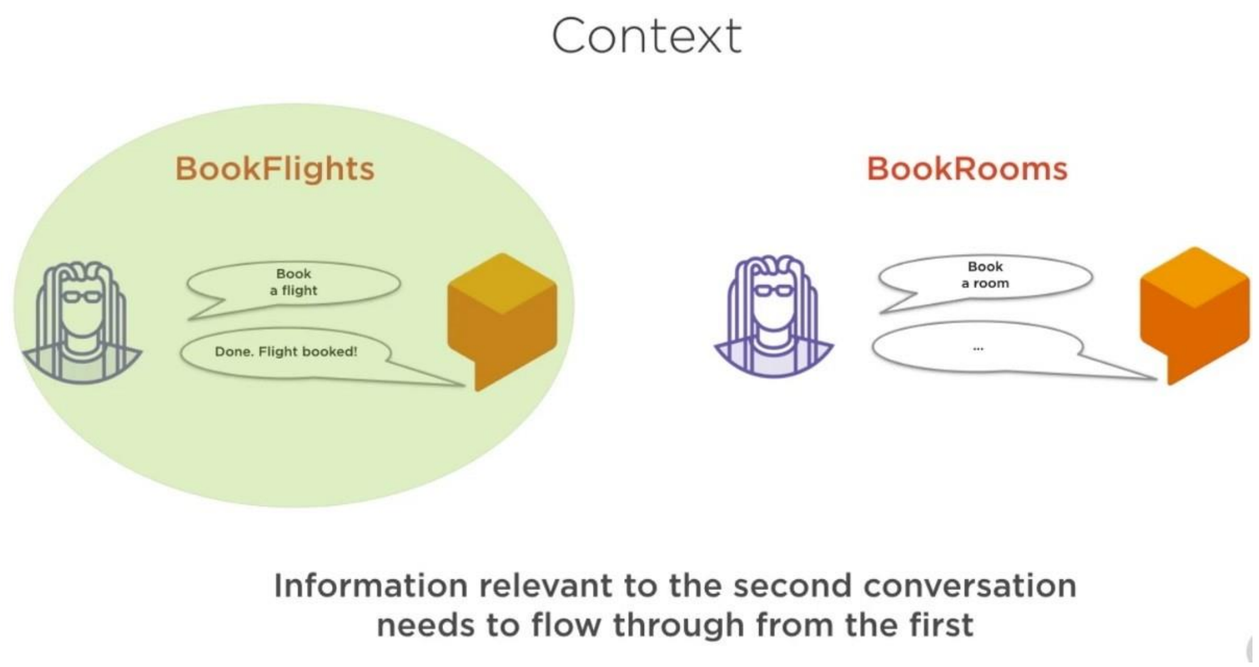
Book a room too → BookRooms intent

.....



Context

These are not 2 separate conversations - the room booking is an extension of the flights conversation

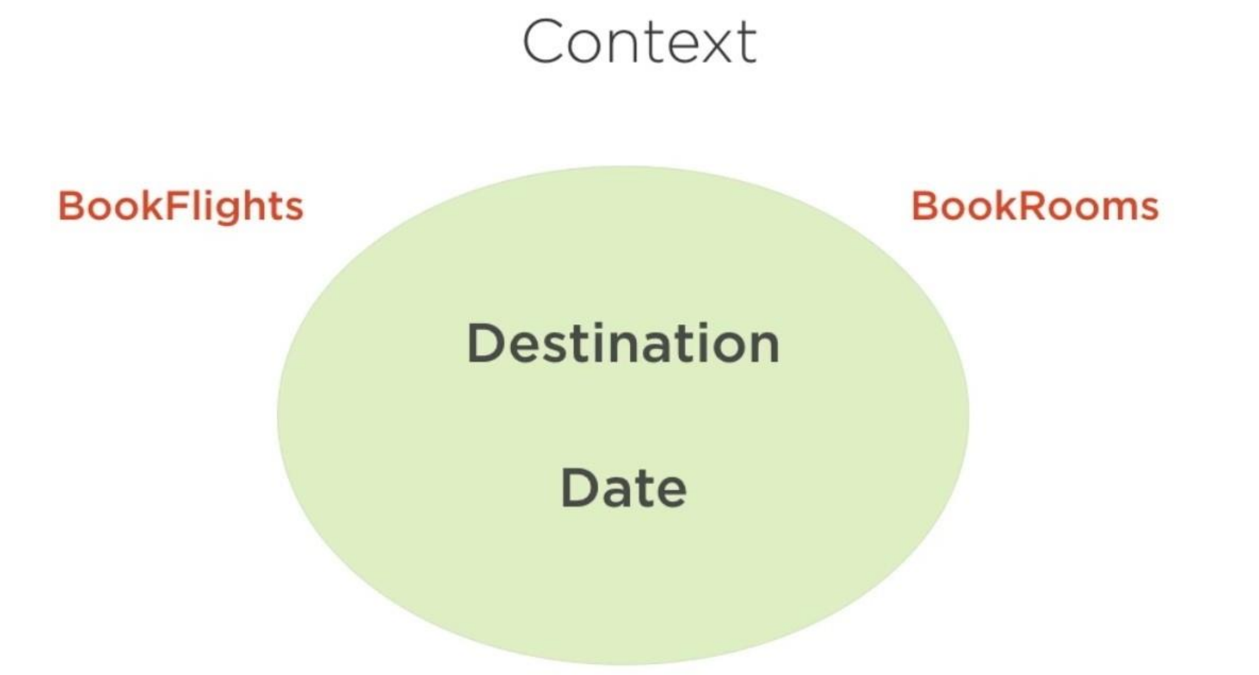


Context

BookFlights

BookRooms

Information relevant to the second conversation needs to flow through from the first



Context

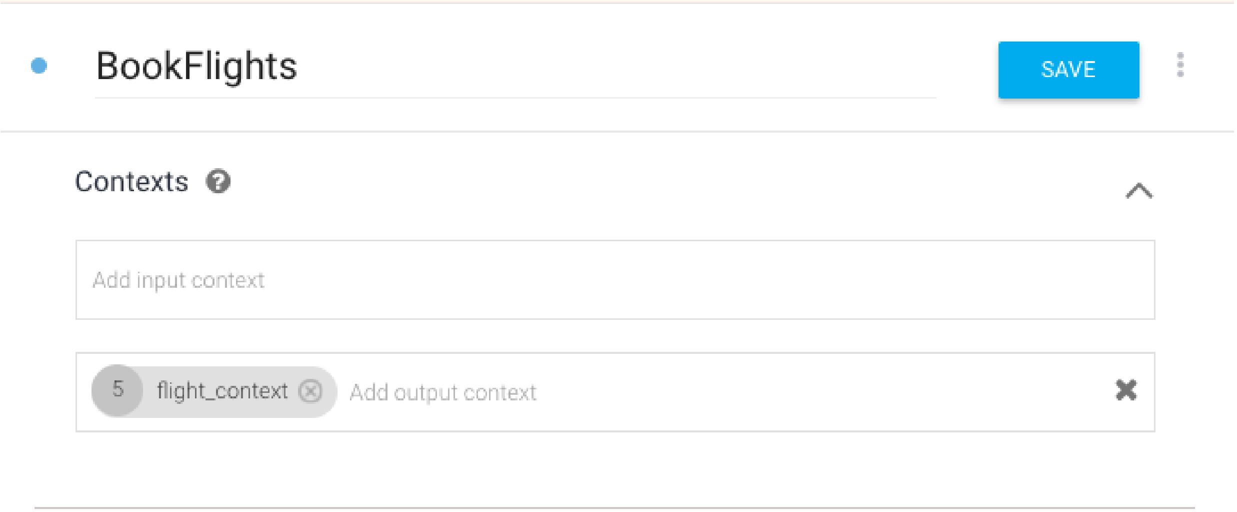
BookFlights

BookRooms

Destination

Date

Now an example of how linear dialogs can flow between multiple intents.



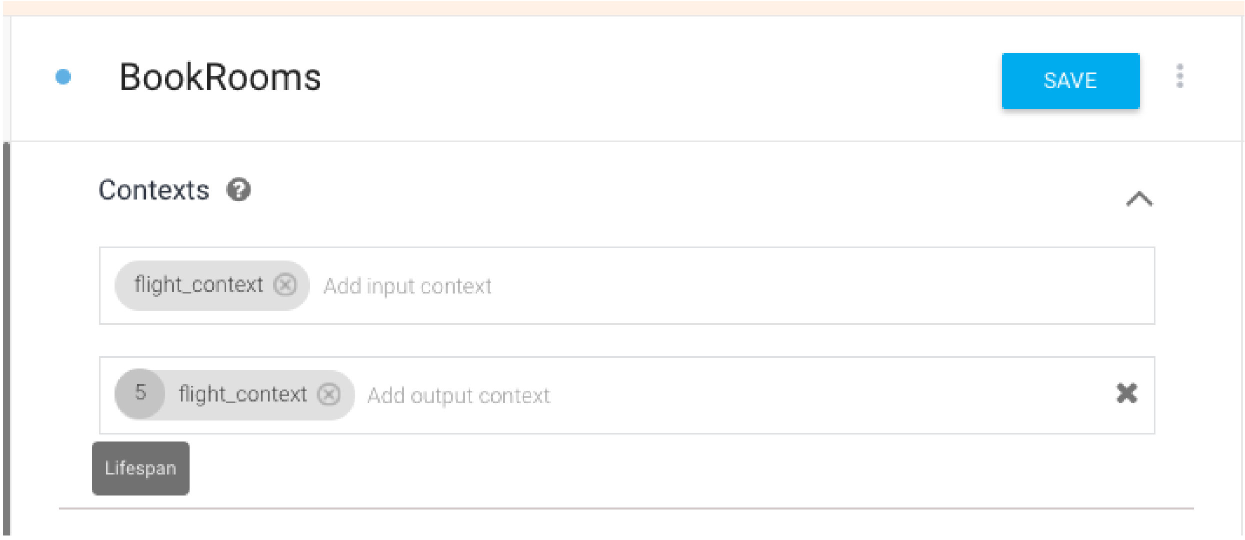
BookFlights

Contexts

Add input context

5 flight_context

Add output context

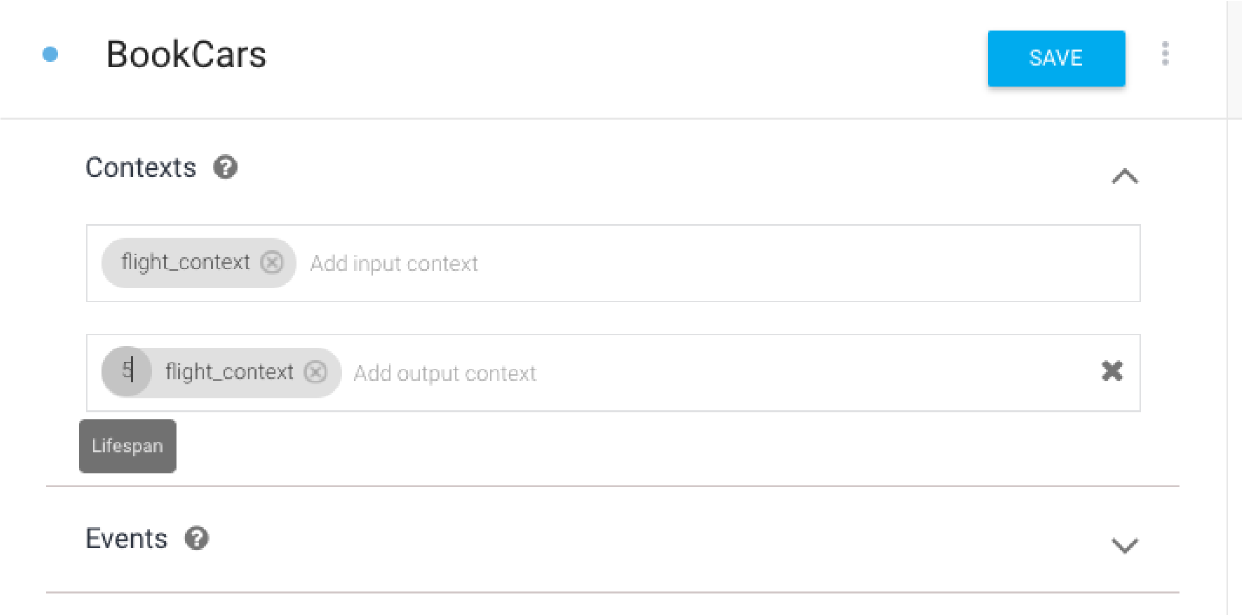


BookRooms

Contexts

flight_context ⊗ Add input context

5 flight_context ⊗ Add output context



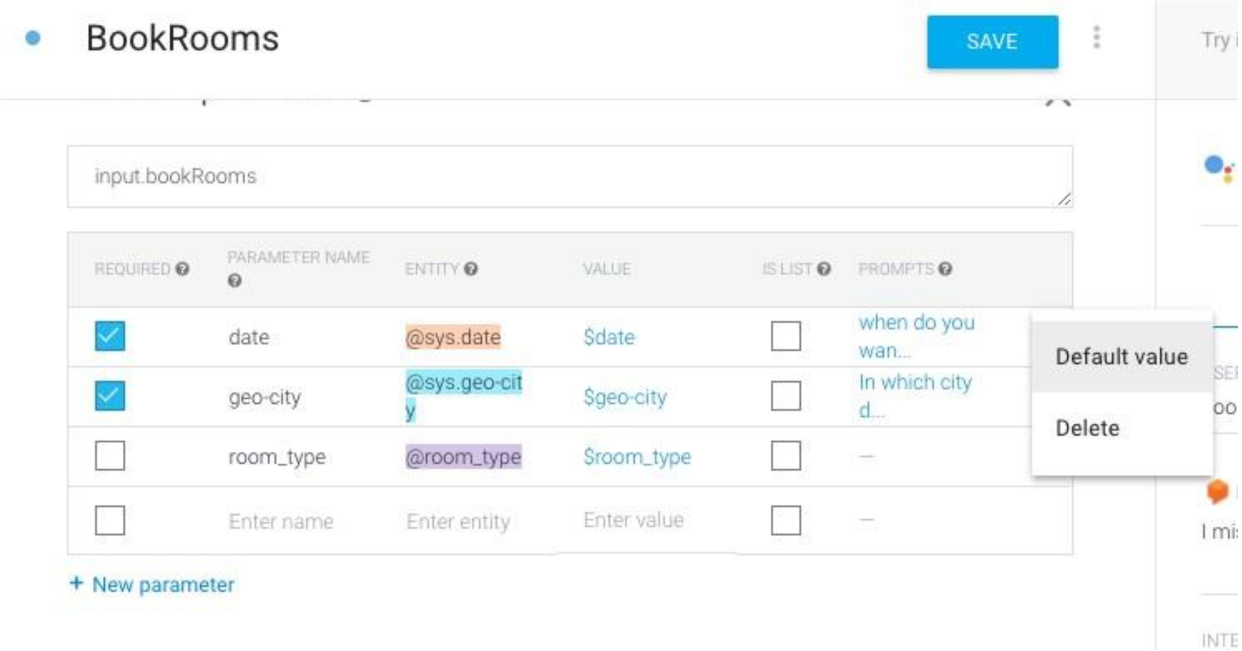
BookCars

Contexts

flight_context ⊗ Add input context

5 flight_context ⊗ Add output context

Events

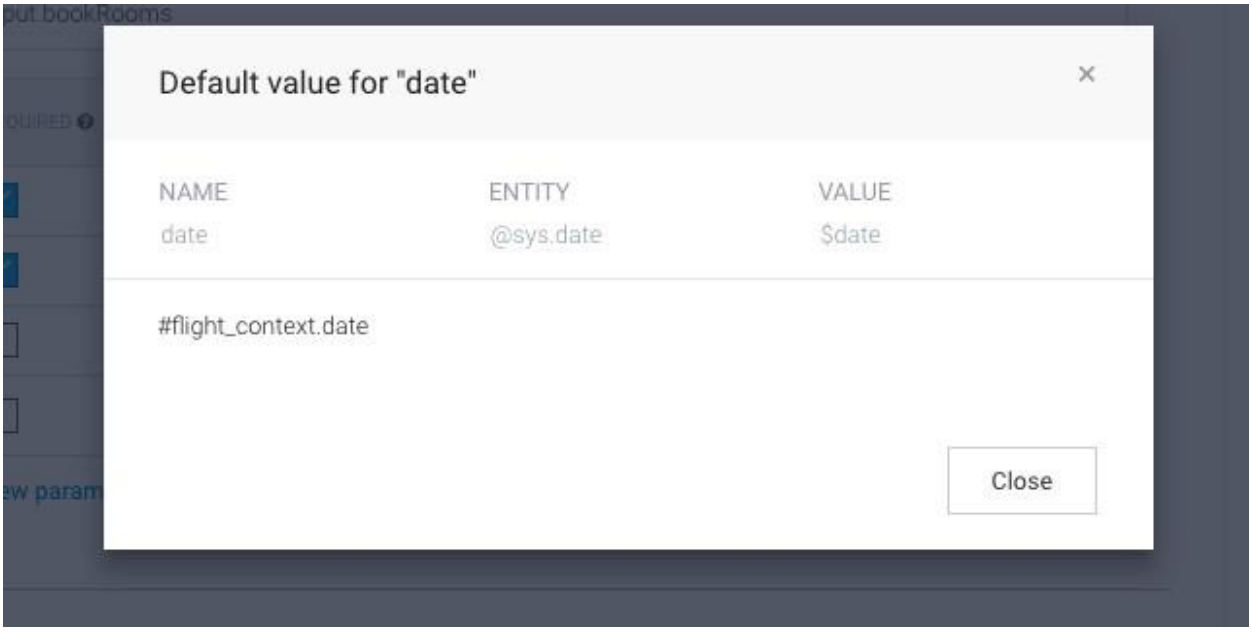


BookRooms

input.bookRooms

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	date	@sys.date	$date	□	when do you wan...
☑	geo-city	@sys.geo-city	$geo-city	□	In which city d...
□	room_type	@room_type	$room_type	□	—
□	Enter name	Enter entity	Enter value	□	—

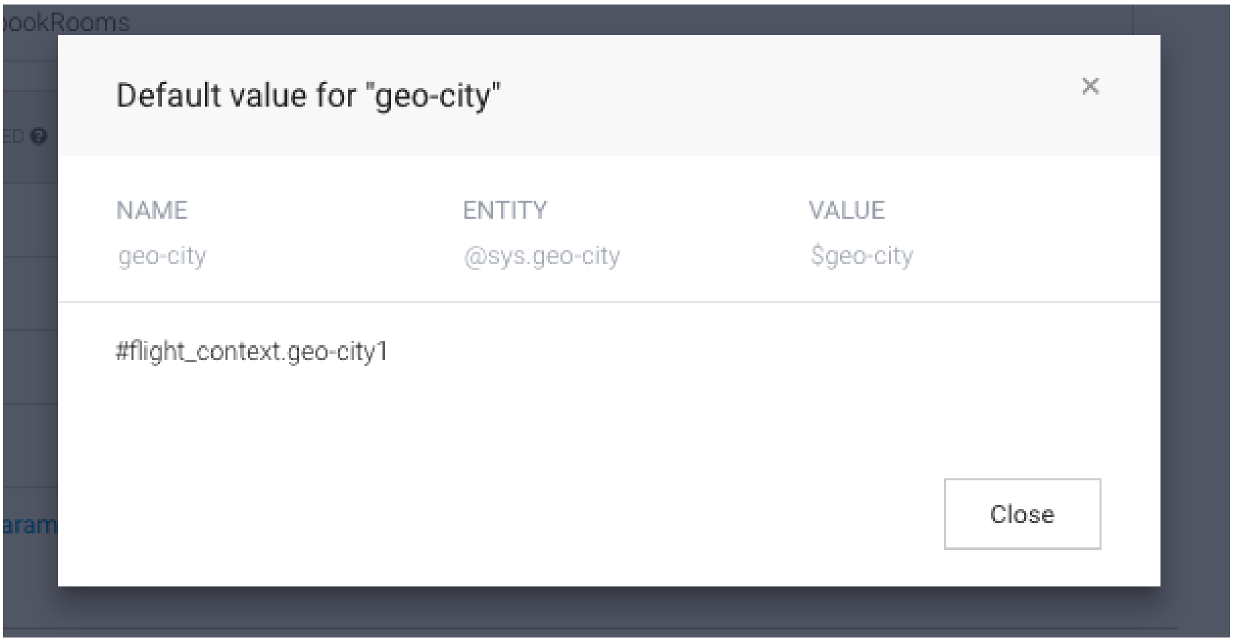
+ New parameter



Default value for "date"

NAME	ENTITY	VALUE
date	@sys.date	$date

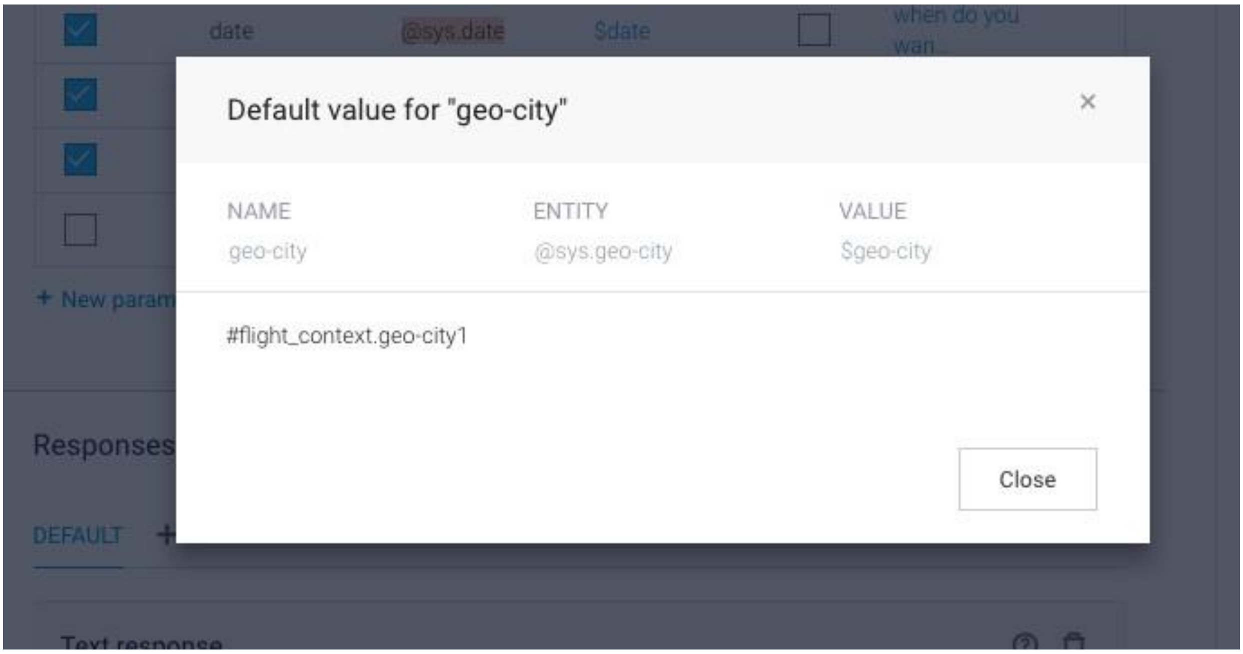
#flight_context.date



Default value for "geo-city"

NAME	ENTITY	VALUE
geo-city	@sys.geo-city	$geo-city

#flight_context.geo-city1



Default value for "geo-city"

NAME	ENTITY	VALUE
geo-city	@sys.geo-city	$geo-city

#flight_context.geo-city1

Within Book Flights intent…create “flight_context” as an output context. the Number 5 represents the lifespan of this context. By default it’s for 5 requests.

Go to BookRooms intent and add “flight_context” as the input context

Go to BookCars intent and add “flight_context” as the input context

Within BookRooms intent..once the input context is it, we gotta assign the default values for parameters.

set deafult value for date as “#flight_context.date”

```
syntax:
#context_name.paramter_name
```

Remember geo-city1 is the destination city…go to BookFlight intent and check out the names.

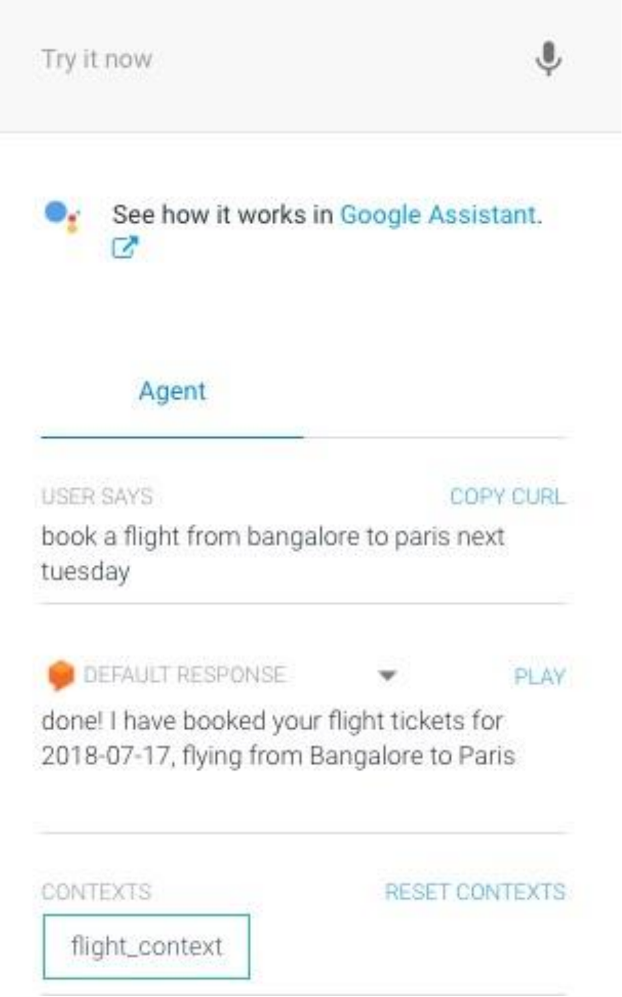
similarly set default values in the bookCars intent as well.

Let’s test out the agent:

1. Book a flight with all necessary inputs.
2. Book a room
3. Book a car (car type alone has to be specified)

Booking a flight with all inputs.

See “flight_context” being setup under contexts!



book a flight from bangalore to paris next tuesday

done! I have booked your flight tickets for 2018-07-17, flying from Bangalore to Paris

flight_context

2. Book a room without specifying any date or city

great, see just saying “book a room” has resulted in a room booked for the destination city and date from the bookFlights intent through the flight_context.

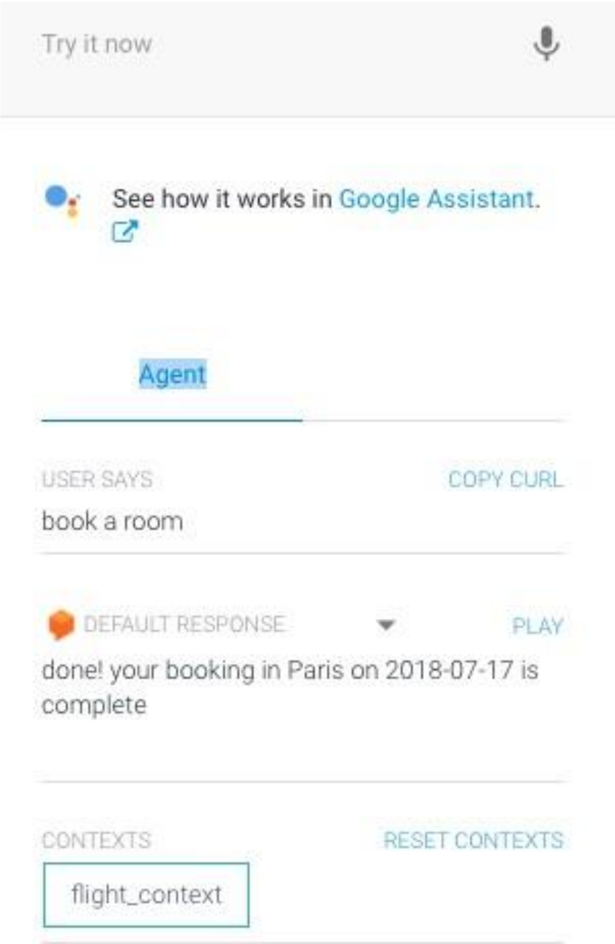
3. Book a car

Book a car — user expression just asks for the type of car

as you can see in the parameter the rest (city, date) are retained from the “flight_context” input context.

Ok, don’t jump off your horses….the issue here is you always need to start with the Bookflight intent…as in, an user has to first book a flight and then a room / car this is not ideal right ?

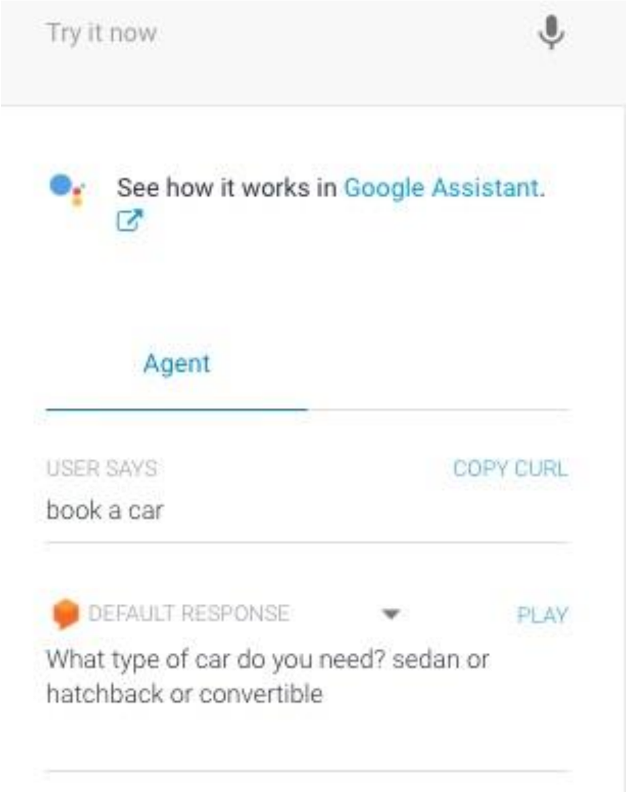
To solve this, we have the **followup intent.**



USER SAYS
book a room

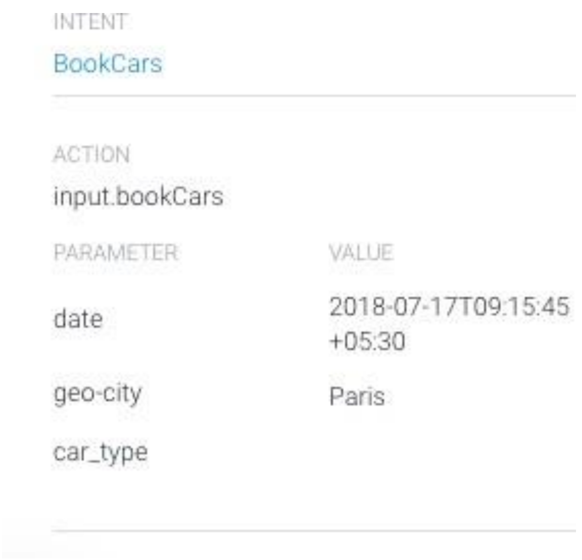
DEFAULT RESPONSE
done! your booking in Paris on 2018-07-17 is complete

CONTEXTS
flight_context



USER SAYS
book a car

DEFAULT RESPONSE
What type of car do you need? sedan or hatchback or convertible



INTENT
BookCars

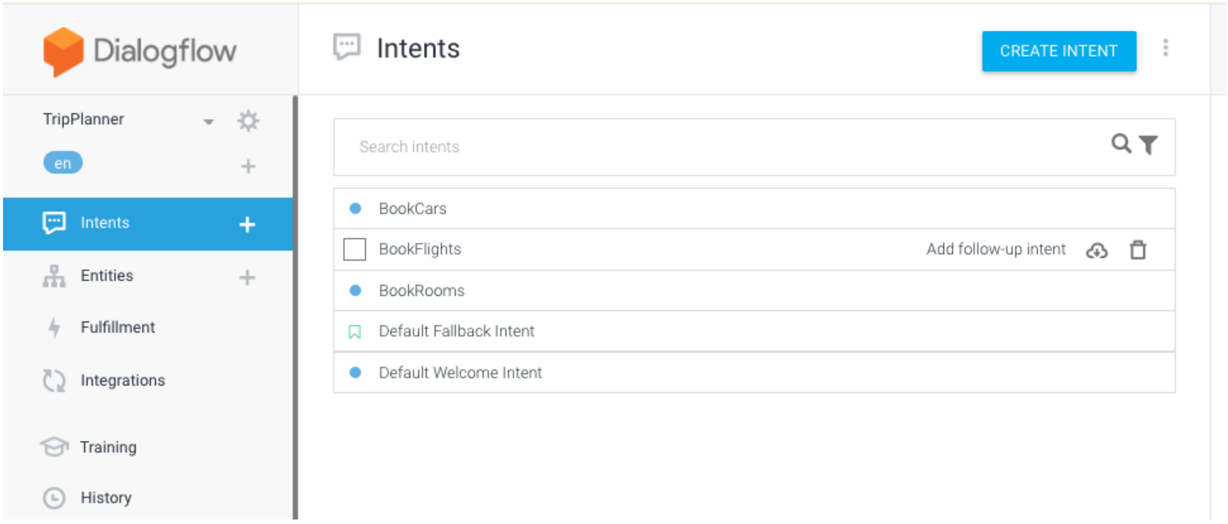
ACTION
input.bookCars

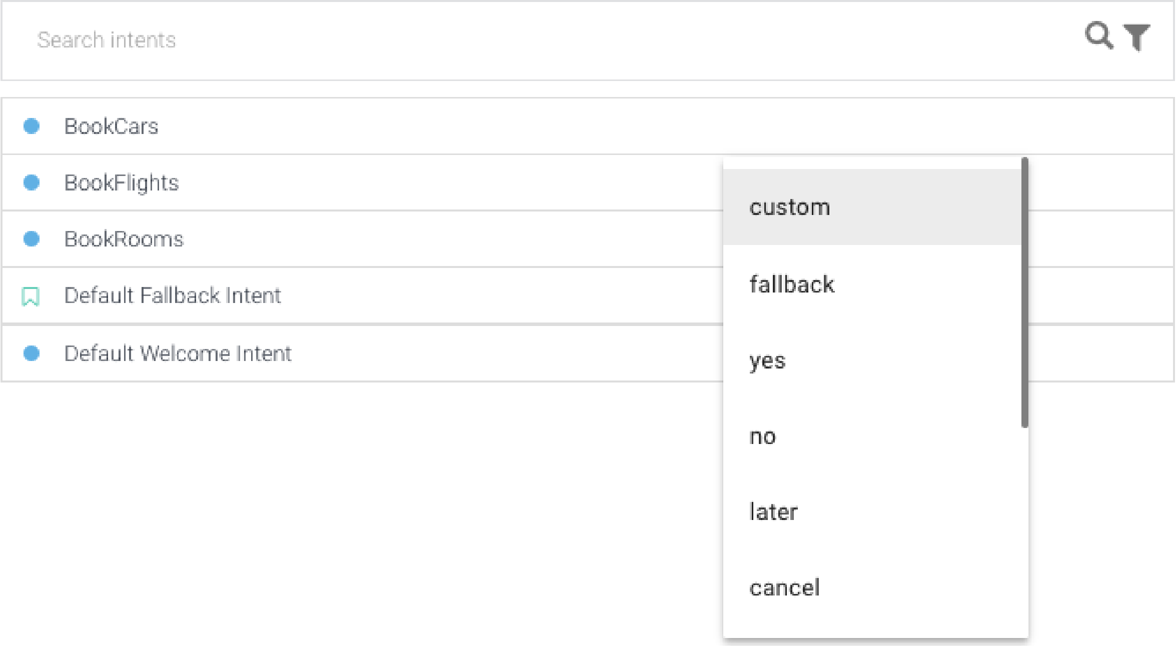
PARAMETER	VALUE
date	2018-07-17T09:15:45 +05:30
geo-city	Paris
car_type	

Linear Dialog using a Followup Intent

1. Clear all configuration settings…the “flight_context” has to be removed and the default values have to be removed form both “bookRooms” & “bookCars” intents. Leave it in the “bookFlights” intent.

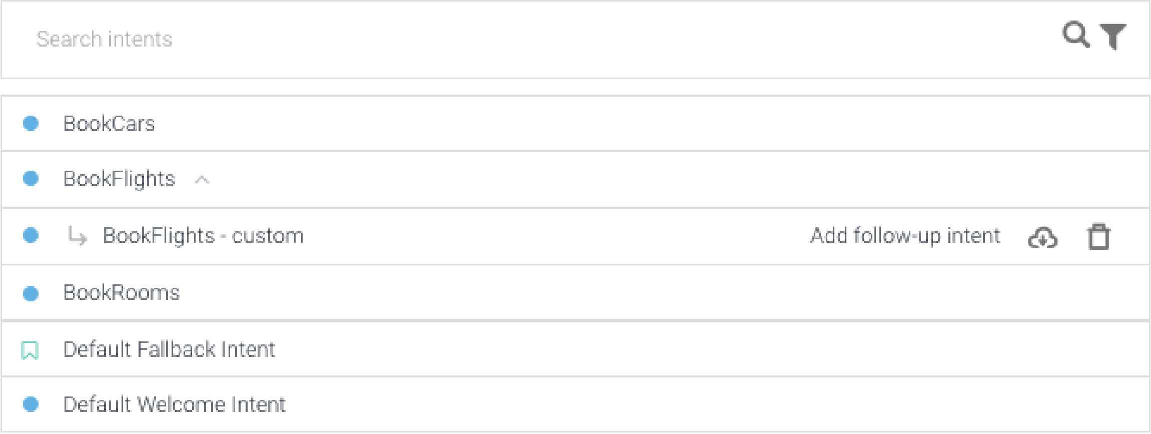
2. Create a custom follow-up intent





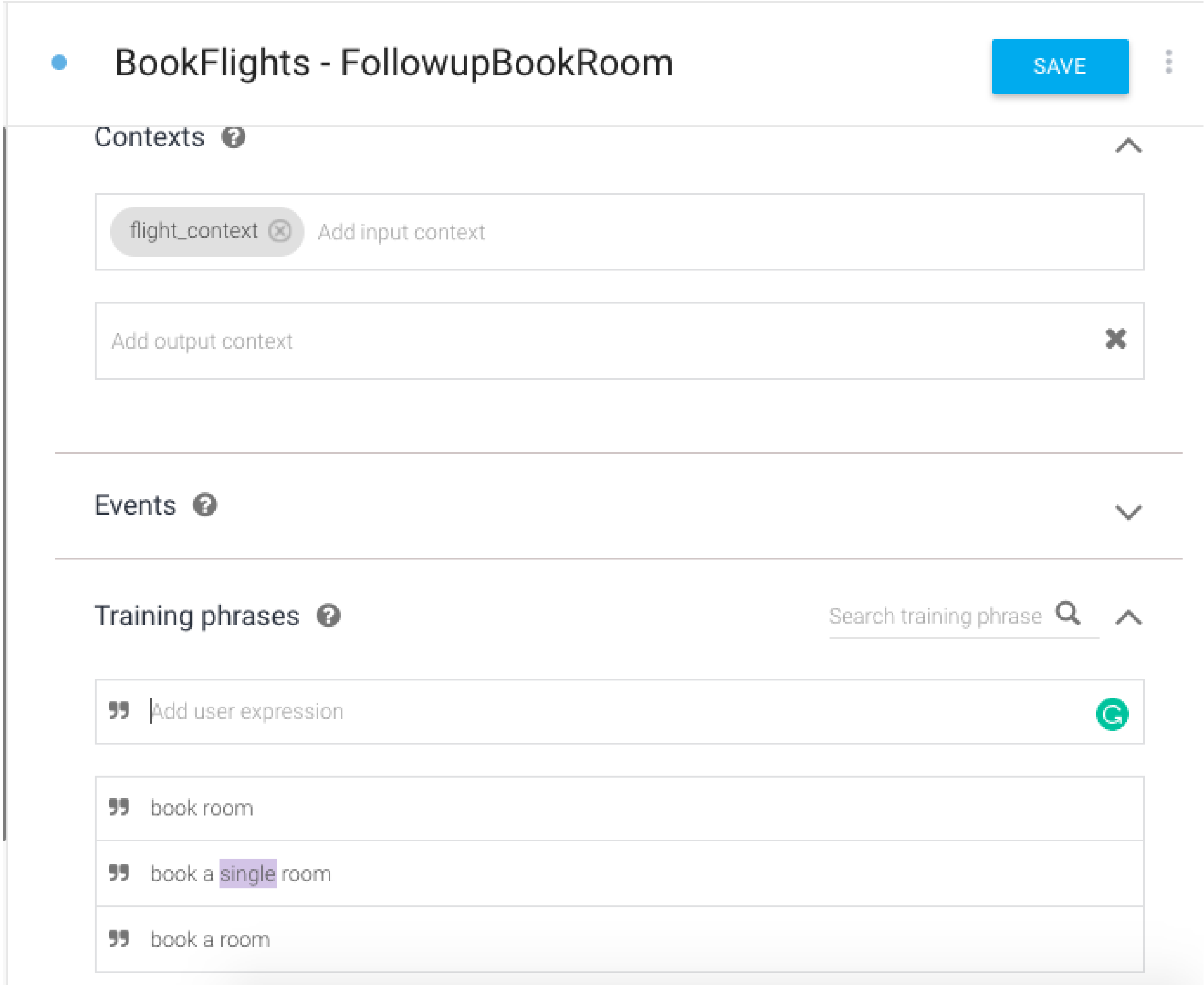
Search intents

- BookCars
- BookFlights
- BookRooms
- Default Fallback Intent
- Default Welcome Intent



Search intents

- BookCars
- BookFlights
- ↳ BookFlights - custom
- BookRooms
- Default Fallback Intent
- Default Welcome Intent



BookFlights - FollowupBookRoom

Contexts

flight_context ⊗ Add input context

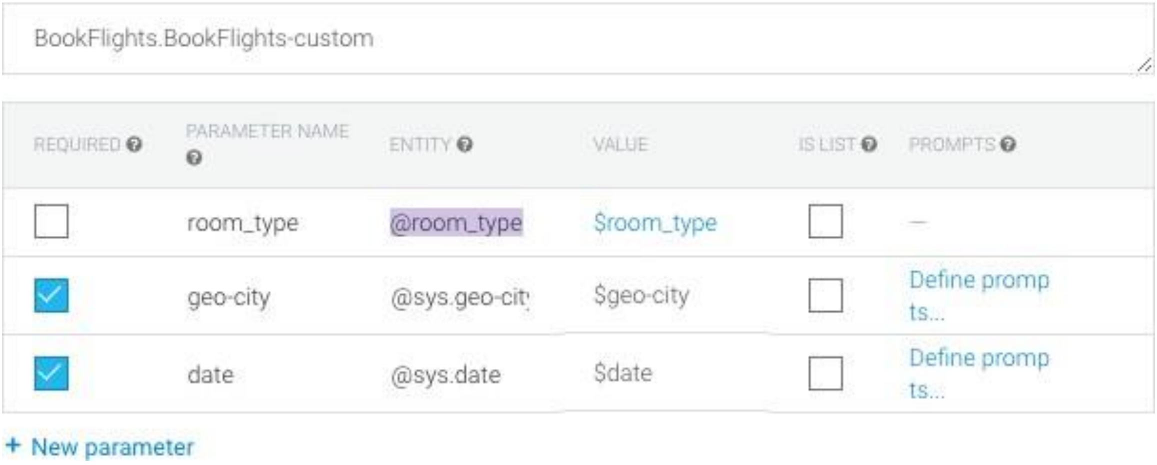
Add output context ✖

Events

Training phrases

Add user expression

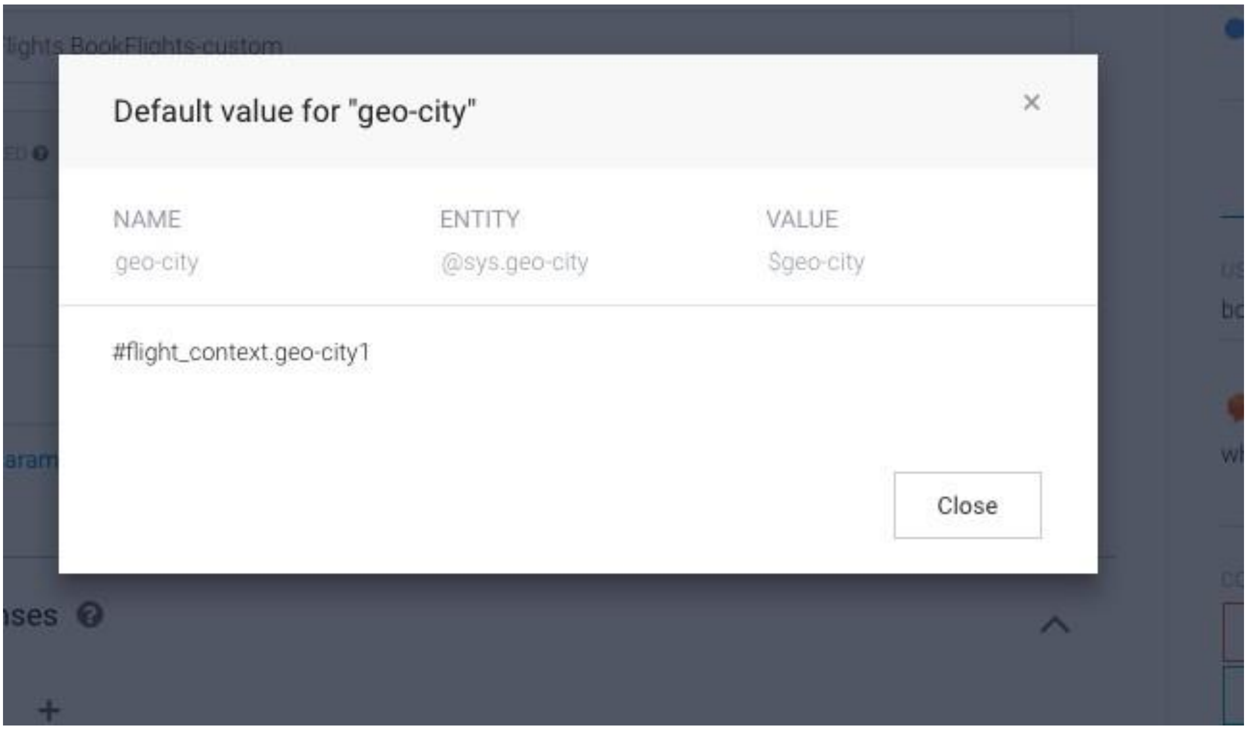
- book room
- book a single room
- book a room



BookFlights.BookFlights-custom

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
□	room_type	@room_type	$room_type	□	–
☑	geo-city	@sys.geo-city	$geo-city	□	Define prompts...
☑	date	@sys.date	$date	□	Define prompts...

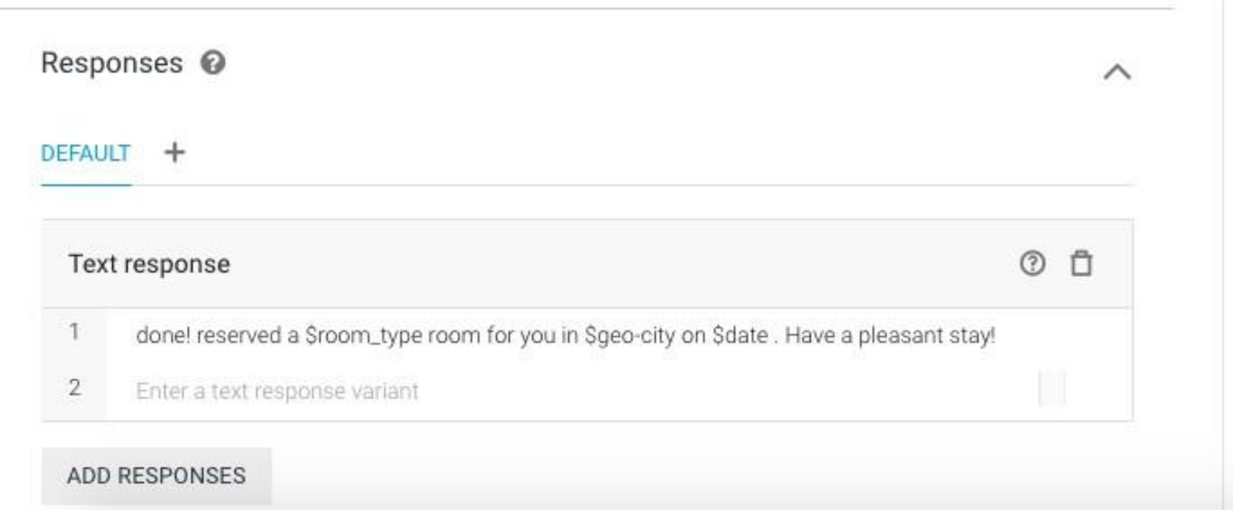
+ New parameter



Default value for "geo-city"

NAME	ENTITY	VALUE
geo-city	@sys.geo-city	$geo-city

#flight_context.geo-city1



Responses

Text response

1 done! reserved a $room_type room for you in $geo-city on $date . Have a pleasant stay!

2 Enter a text response variant

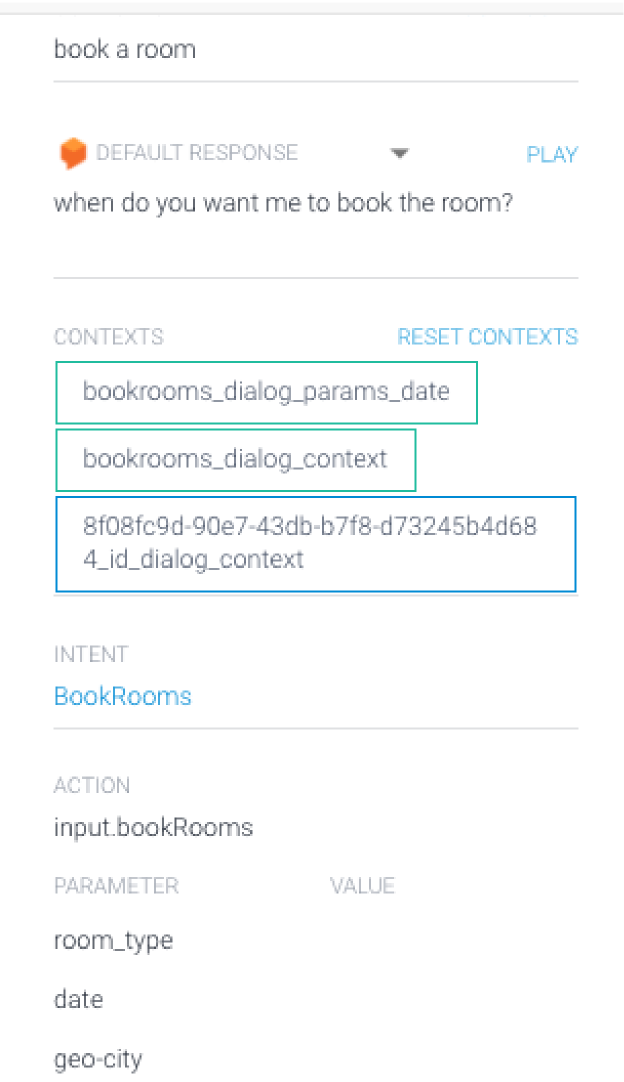
Set the input context as flight_context which is the output context of bookFlights intent, remember?

set few user expressions too…

Make sure you manually add the geocity &date parameters

set the default response too…

So a normal book a room would trigger the BookRooms intent.. see below booking a flight and then using the same expression would trigger the followup intent



book a room

when do you want me to book the room?

bookrooms_dialog_params_date

bookrooms_dialog_context

8f08fc9d-90e7-43db-b7f8-d73245b4d684_id_dialog_context

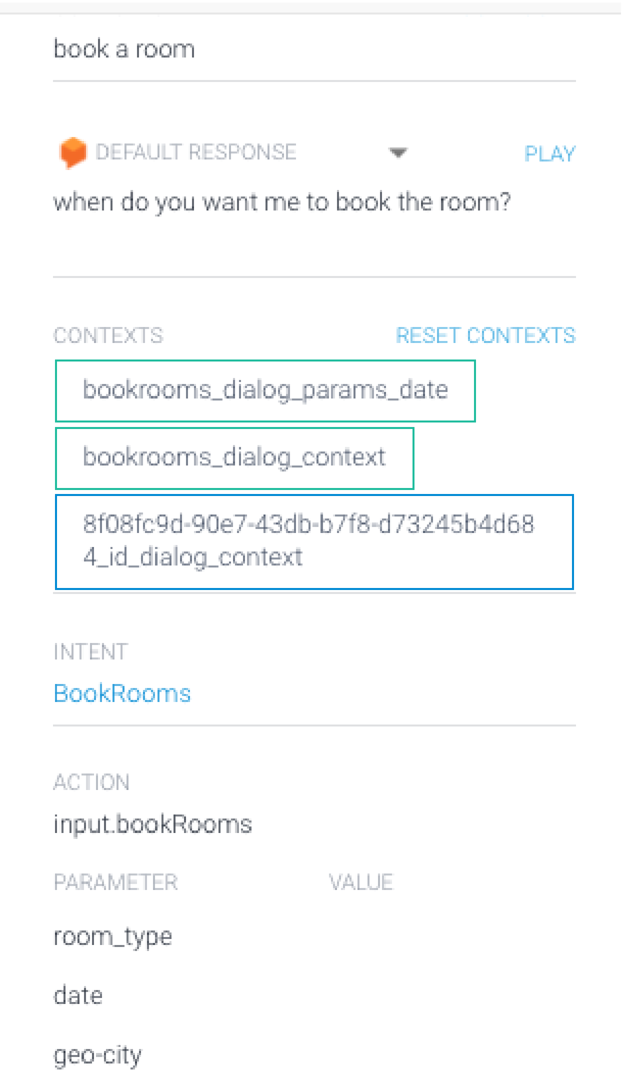
INTENT

BookRooms

ACTION

input.bookRooms

PARAMETER	VALUE
room_type	
date	
geo-city	



book a room

DEFAULT RESPONSE

when do you want me to book the room?

CONTEXTS

bookrooms_dialog_params_date

bookrooms_dialog_context

8f08fc9d-90e7-43db-b7f8-d73245b4d684_id_dialog_context

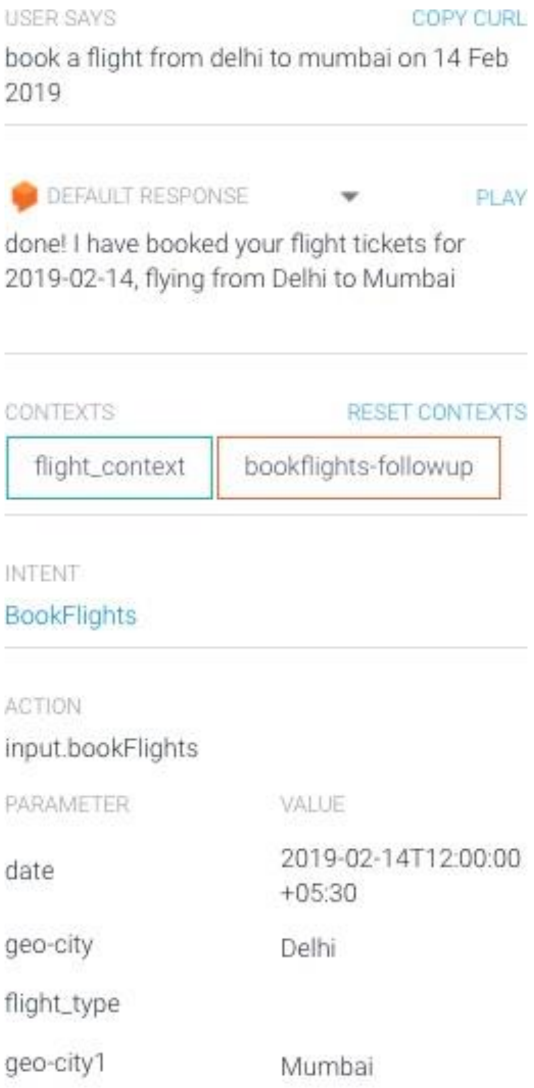
INTENT

BookRooms

ACTION

input.bookRooms

PARAMETER	VALUE
room_type	
date	
geo-city	



USER SAYS

book a flight from delhi to mumbai on 14 Feb 2019

DEFAULT RESPONSE

done! I have booked your flight tickets for 2019-02-14, flying from Delhi to Mumbai

CONTEXTS

flight_context bookflights-followup

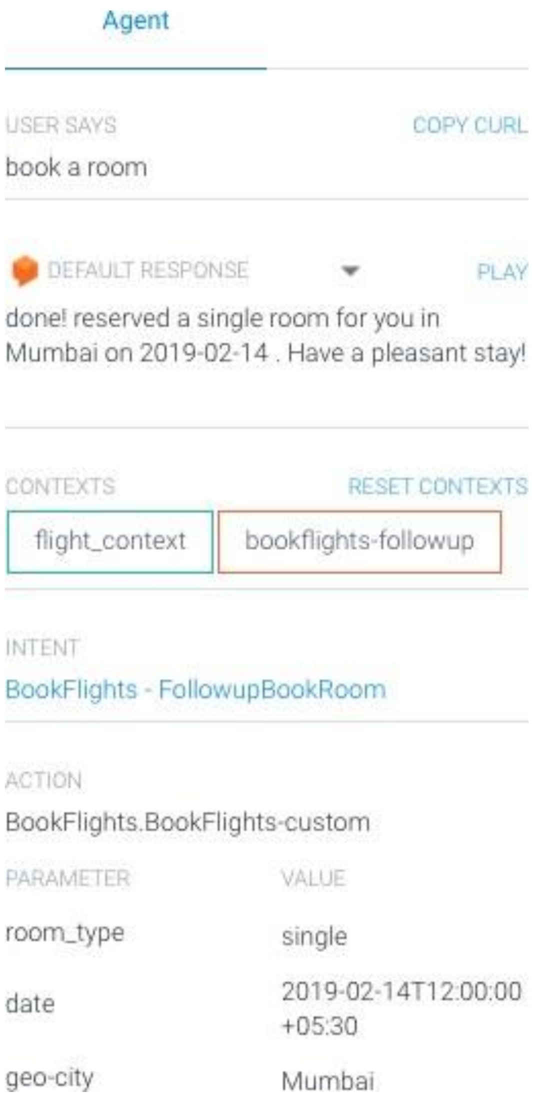
INTENT

BookFlights

ACTION

input.bookFlights

PARAMETER	VALUE
date	2019-02-14T12:00:00 +05:30
geo-city	Delhi
flight_type	
geo-city1	Mumbai



USER SAYS

book a room

DEFAULT RESPONSE

done! reserved a single room for you in Mumbai on 2019-02-14 . Have a pleasant stay!

CONTEXTS

flight_context bookflights-followup

INTENT

BookFlights - FollowupBookRoom

ACTION

BookFlights.BookFlights-custom

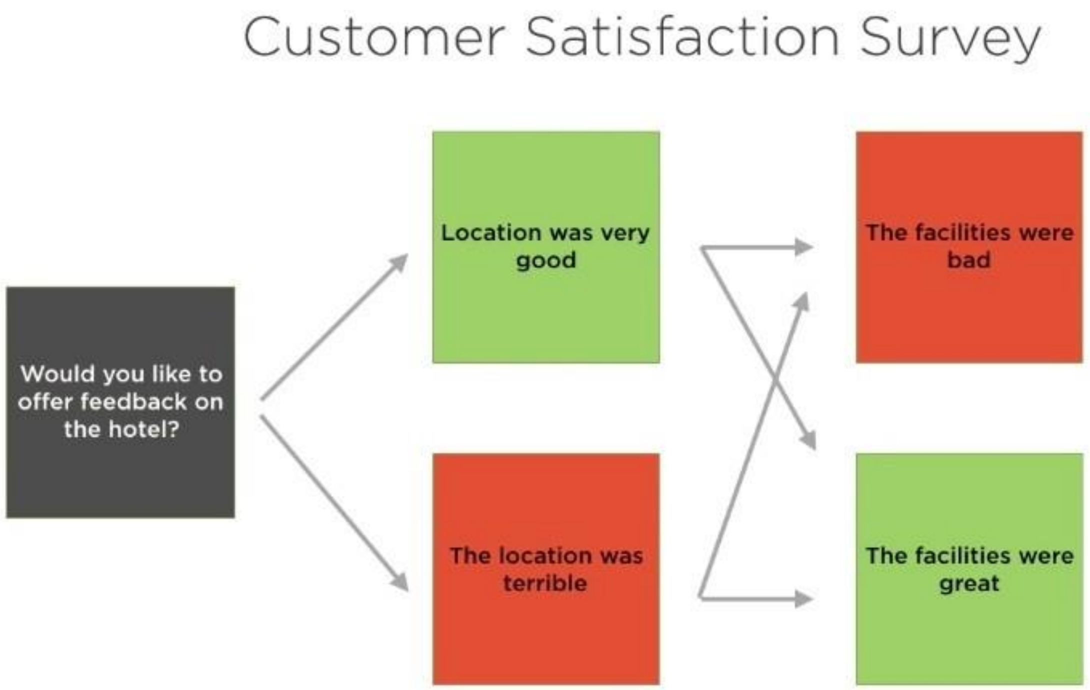
PARAMETER	VALUE
room_type	single
date	2019-02-14T12:00:00 +05:30
geo-city	Mumbai

Till now we have seen only linear dialogs…but real-world conversations aren't linear…don't trust me? try getting into an argument with your girl :P

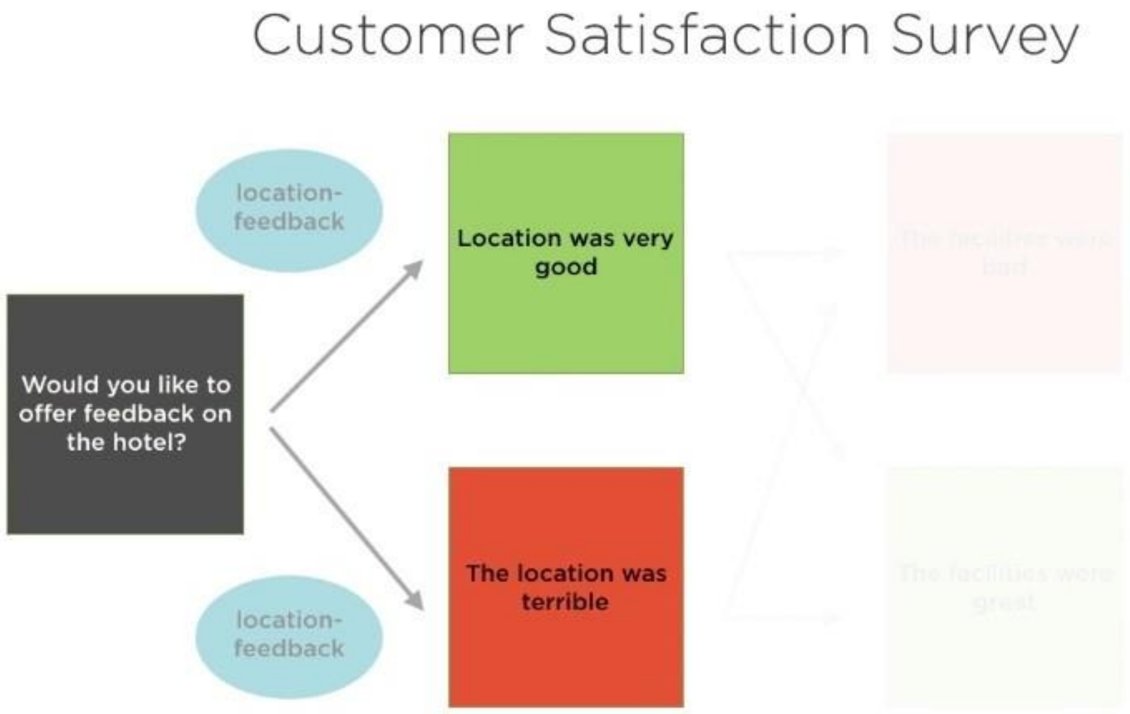
Non-linear dialogs branch to the next intent based on responses from the previous intent,

eg:

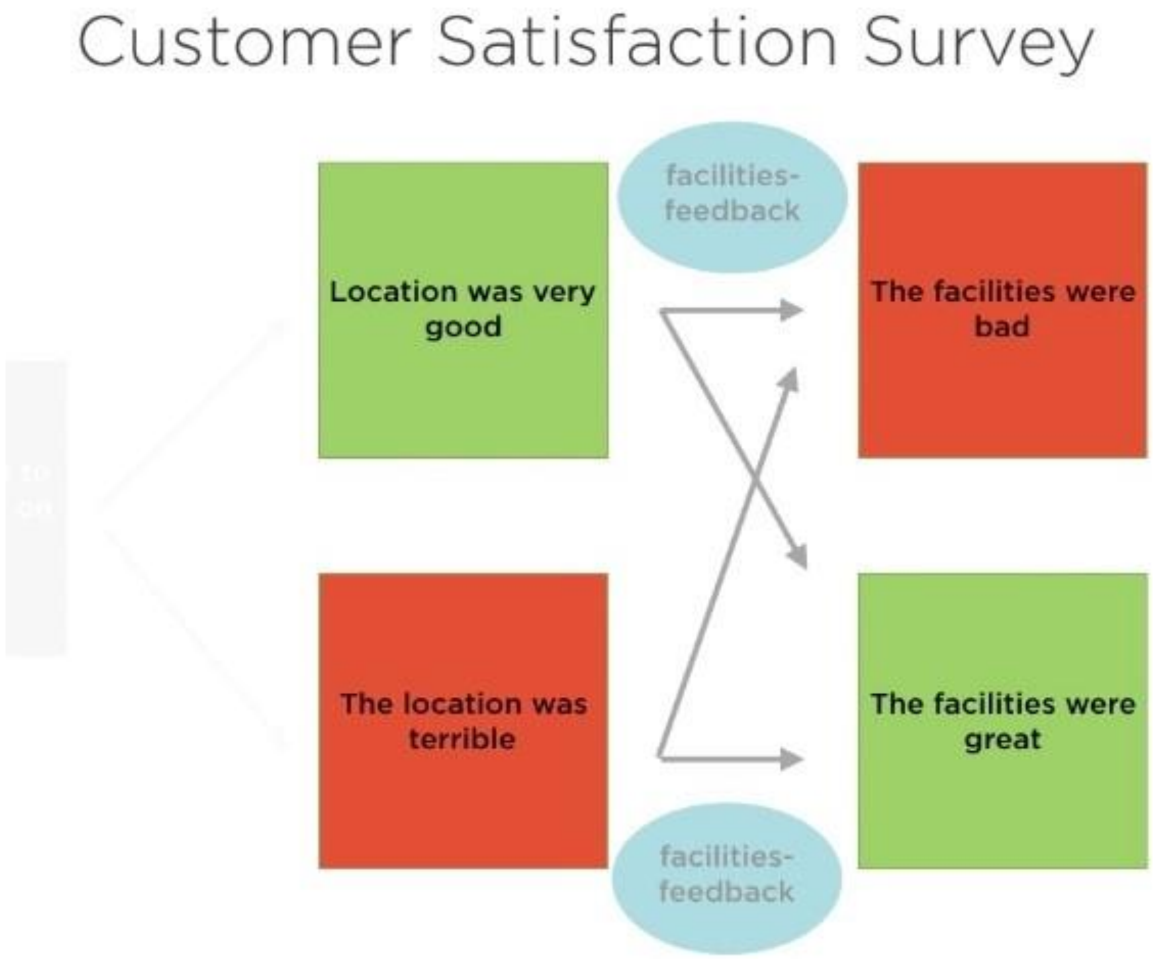
customer satisfaction survey


Customer Satisfaction Survey
Would you like to offer feedback on the hotel?
Location was very good
The location was terrible
The facilities were bad
The facilities were great

Location Feedback:


Customer Satisfaction Survey
location-feedback
location-feedback
Would you like to offer feedback on the hotel?
Location was very good
The location was terrible
The facilities were bad
The facilities were great

Facilities Feedback



Customer Satisfaction Survey

Location was very good

facilities-feedback

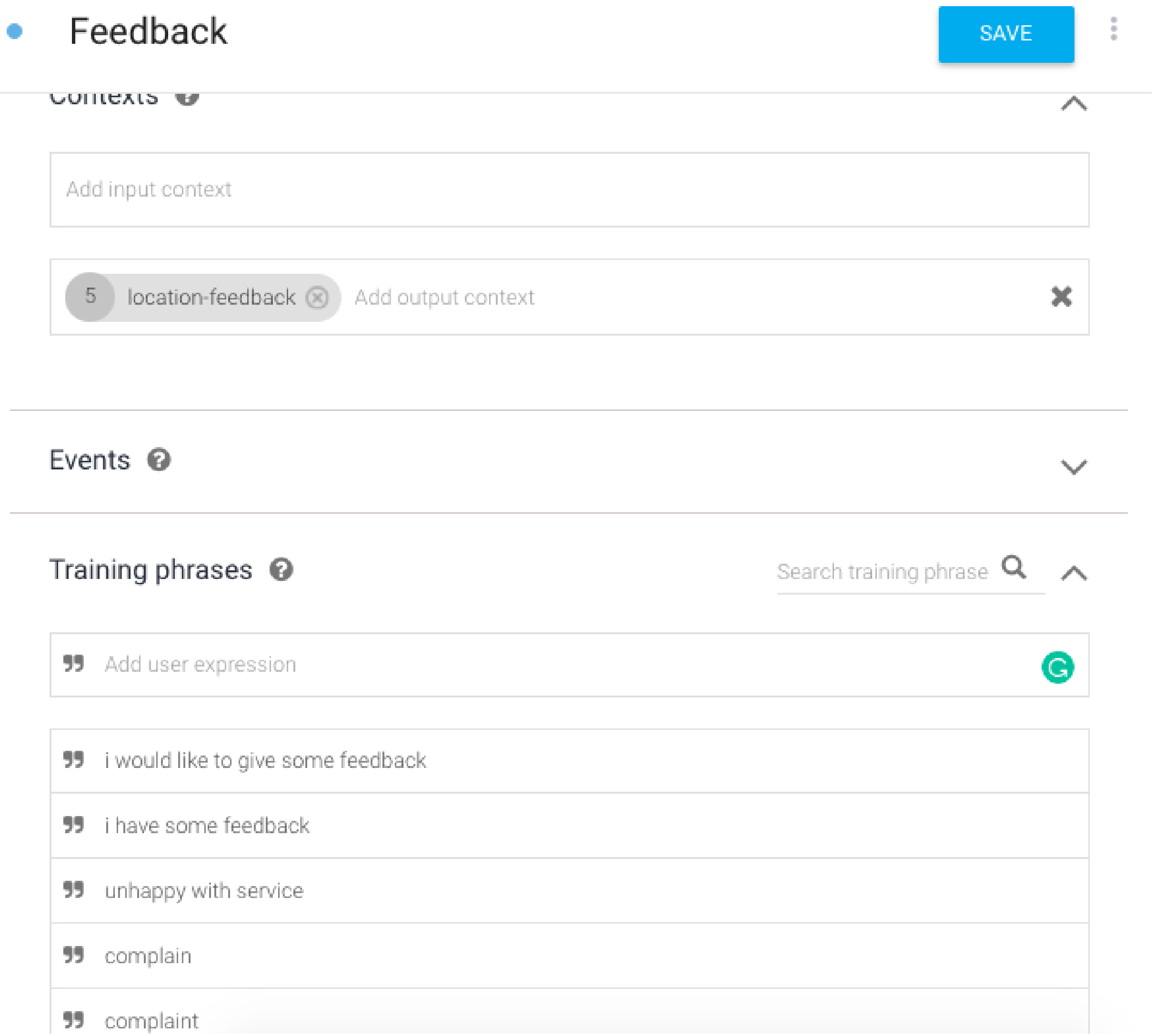
The facilities were bad

The location was terrible

facilities-feedback

The facilities were great

Configuration: create a new **feedback intent**—set output context, few training phrases and the text reponse



Feedback

SAVE

Contexts

Add input context

5 location-feedback

Add output context

Events

Training phrases

Search training phrase

Add user expression

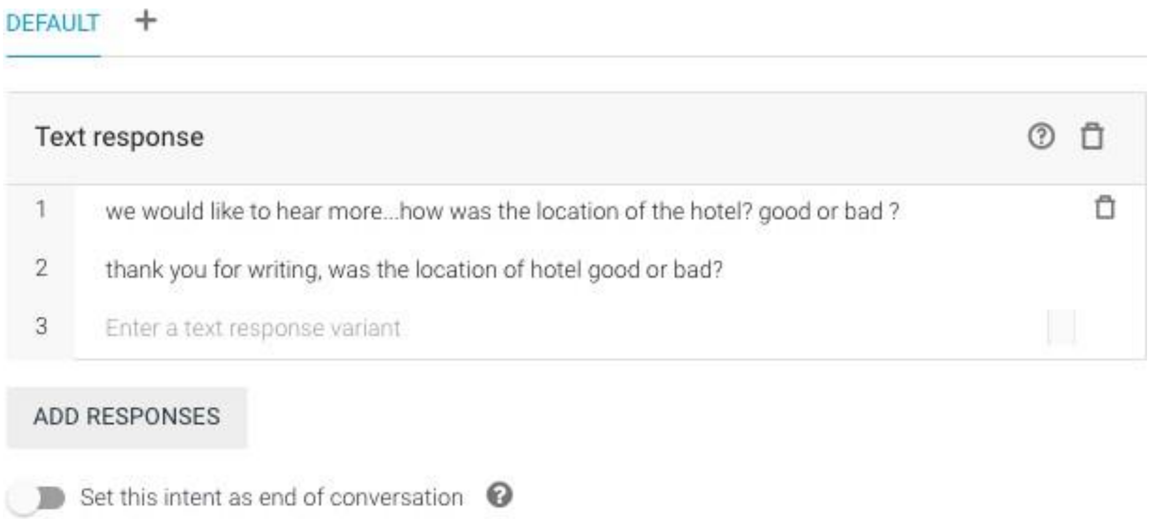
i would like to give some feedback

i have some feedback

unhappy with service

complain

complaint



DEFAULT +

Text response

1 we would like to hear more...how was the location of the hotel? good or bad ?

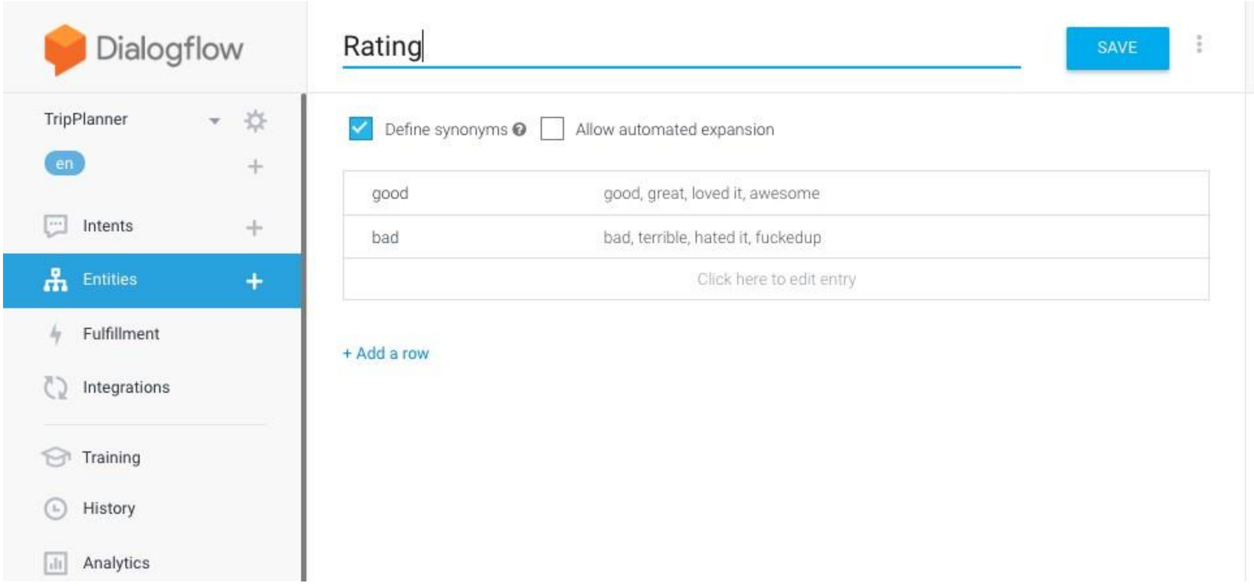
2 thank you for writing, was the location of hotel good or bad?

3 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

Set a new entity called rating



Rating

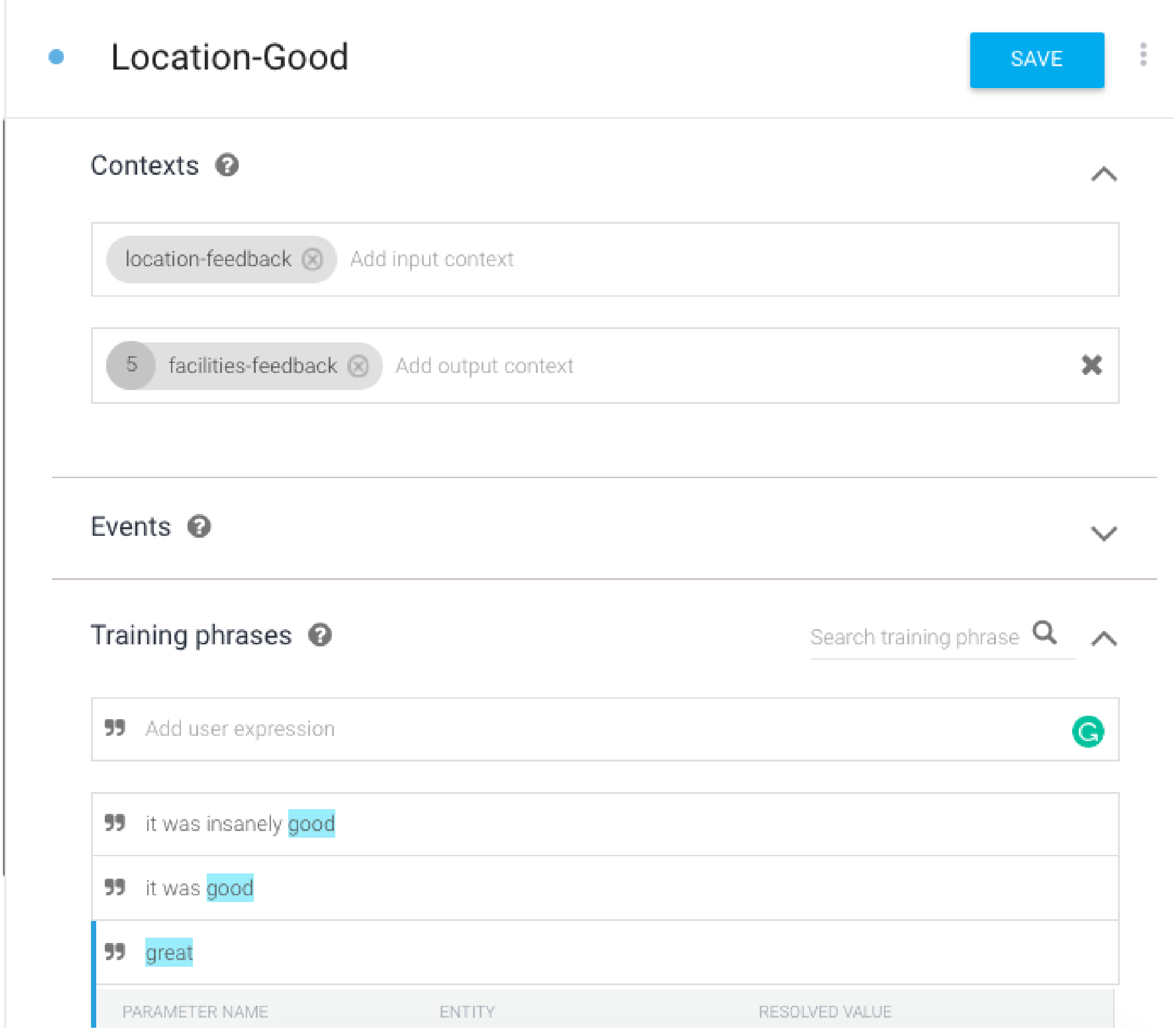
☑ Define synonyms ❓ □ Allow automated expansion

good	good, great, loved it, awesome
bad	bad, terrible, hated it, fuckedup

Click here to edit entry

+ Add a row

New Intent: Location- Good



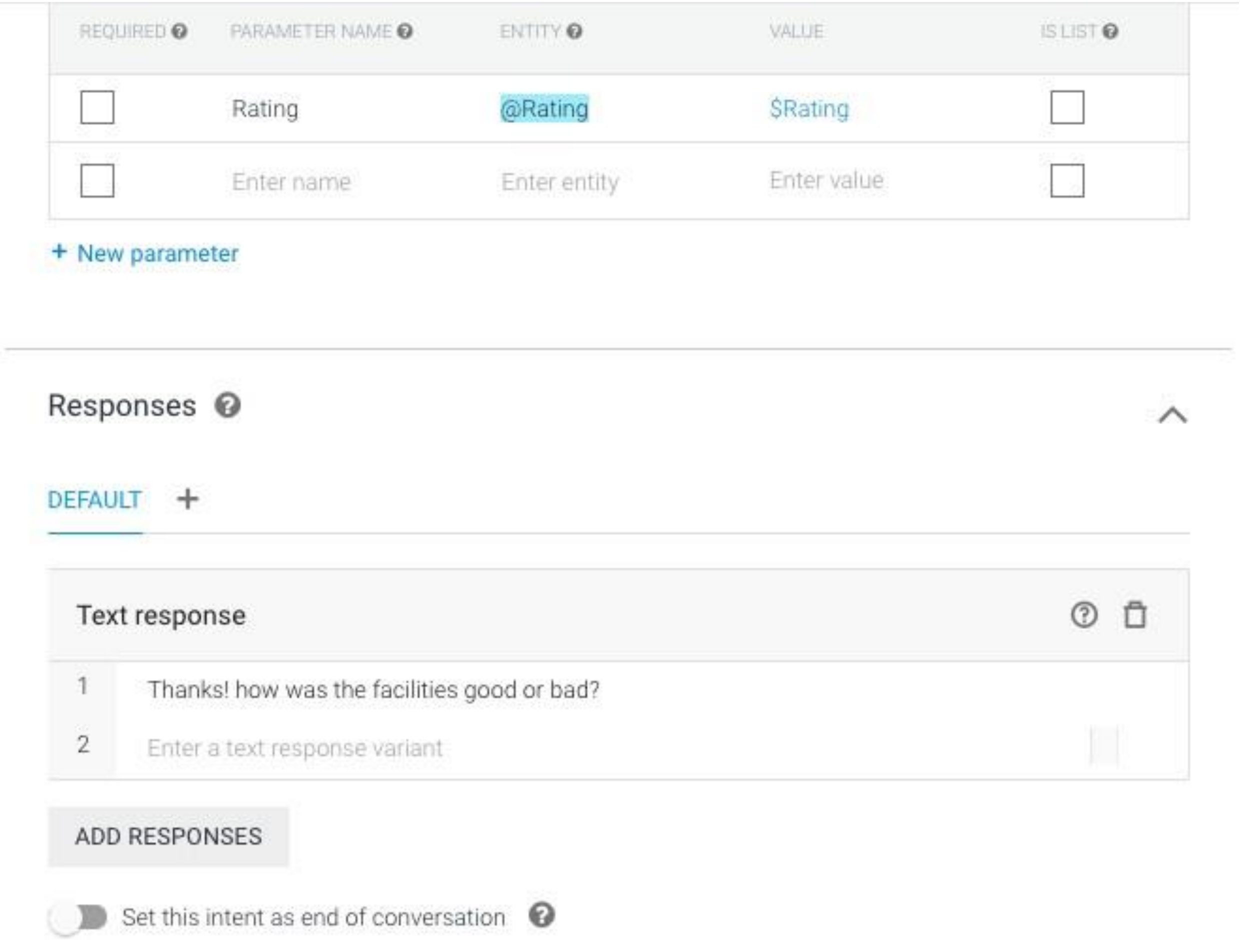
Location-Good

location-feedback ⊗ Add input context

5 facilities-feedback ⊗ Add output context ✖

Add user expression

- it was insanely good
- it was good
- great



REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	Rating	@Rating	$Rating	□
□	Enter name	Enter entity	Enter value	□

+ New parameter

Responses

DEFAULT +

Text response

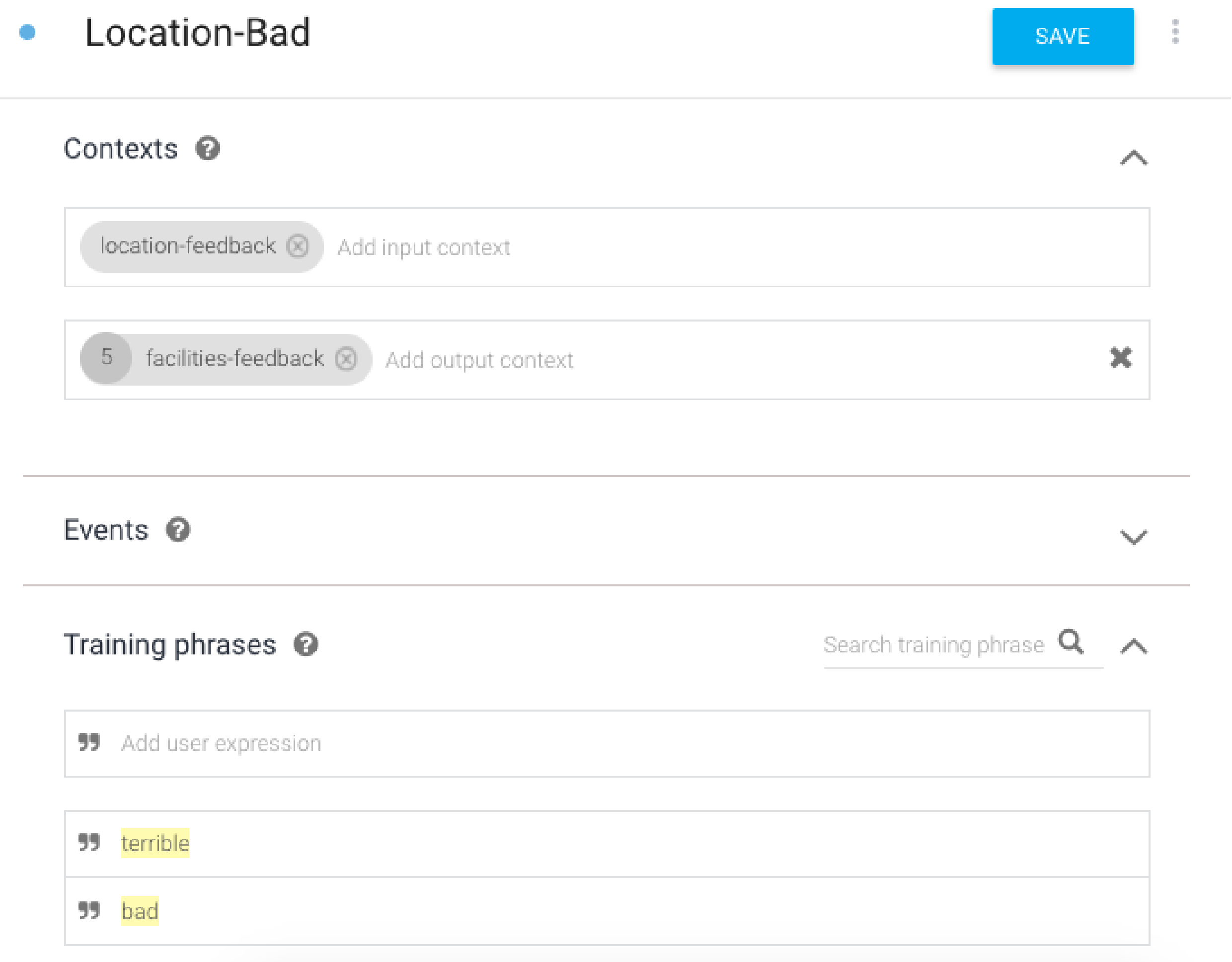
1 Thanks! how was the facilities good or bad?

2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

New Intent : Location—Bad



Location-Bad

Contexts

location-feedback ⊗ Add input context

5 facilities-feedback ⊗ Add output context ✖

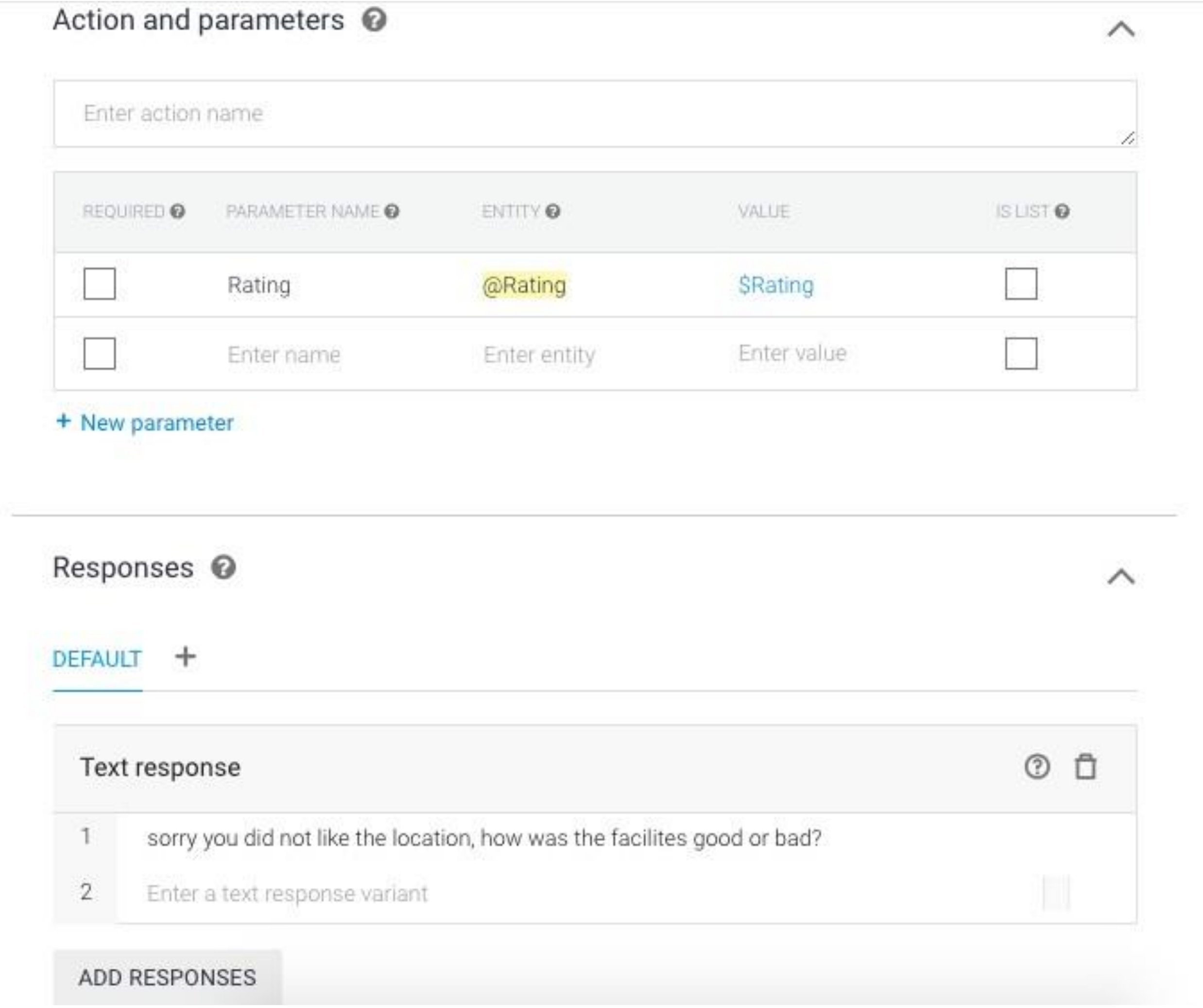
Events

Training phrases

Add user expression

terrible

bad

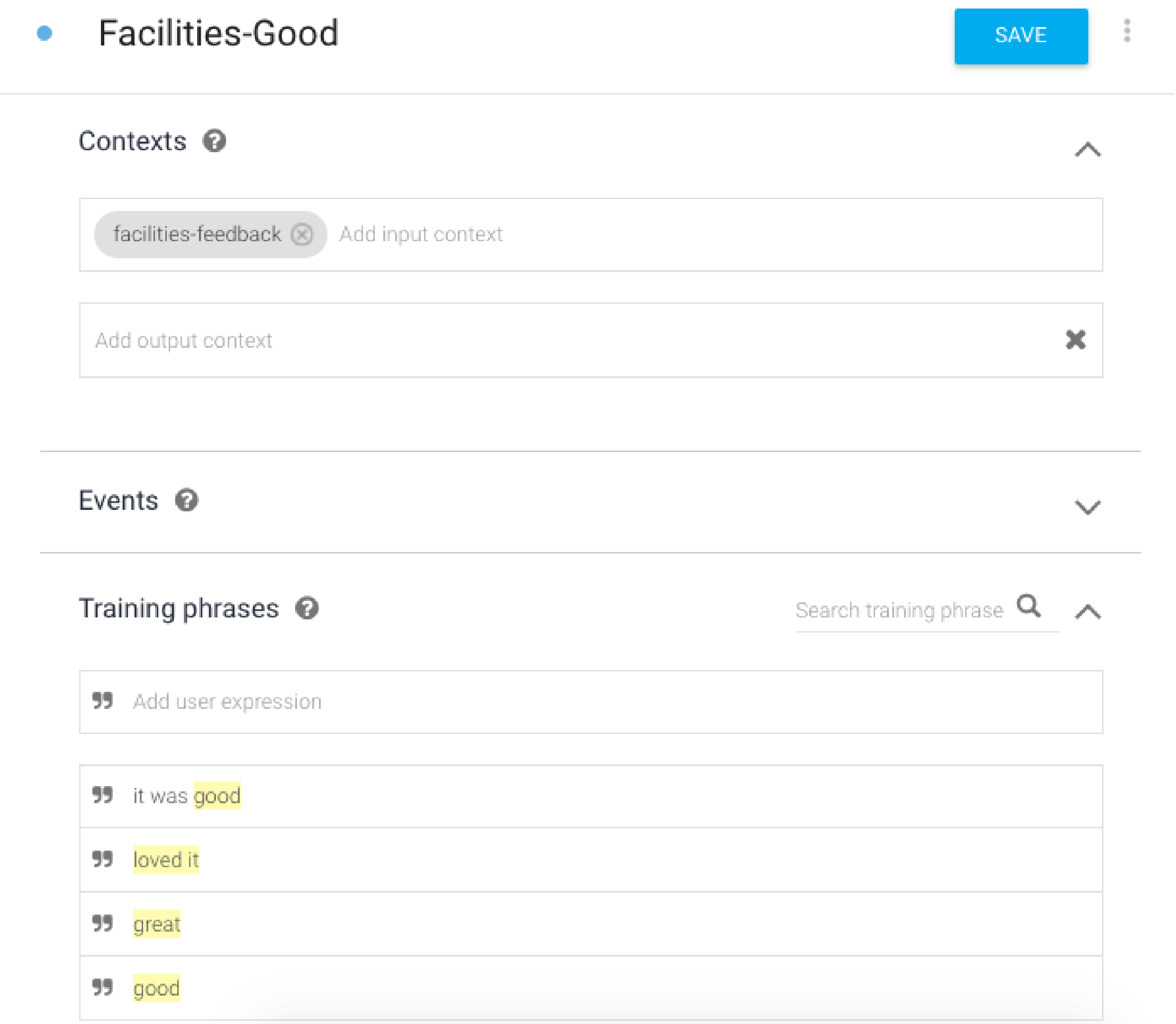


REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	Rating	@Rating	$Rating	□
□	Enter name	Enter entity	Enter value	□

1 sorry you did not like the location, how was the facilites good or bad?

2 Enter a text response variant

New Intent: Facilitites—Good



Facilities-Good

Contexts

facilities-feedback

Events

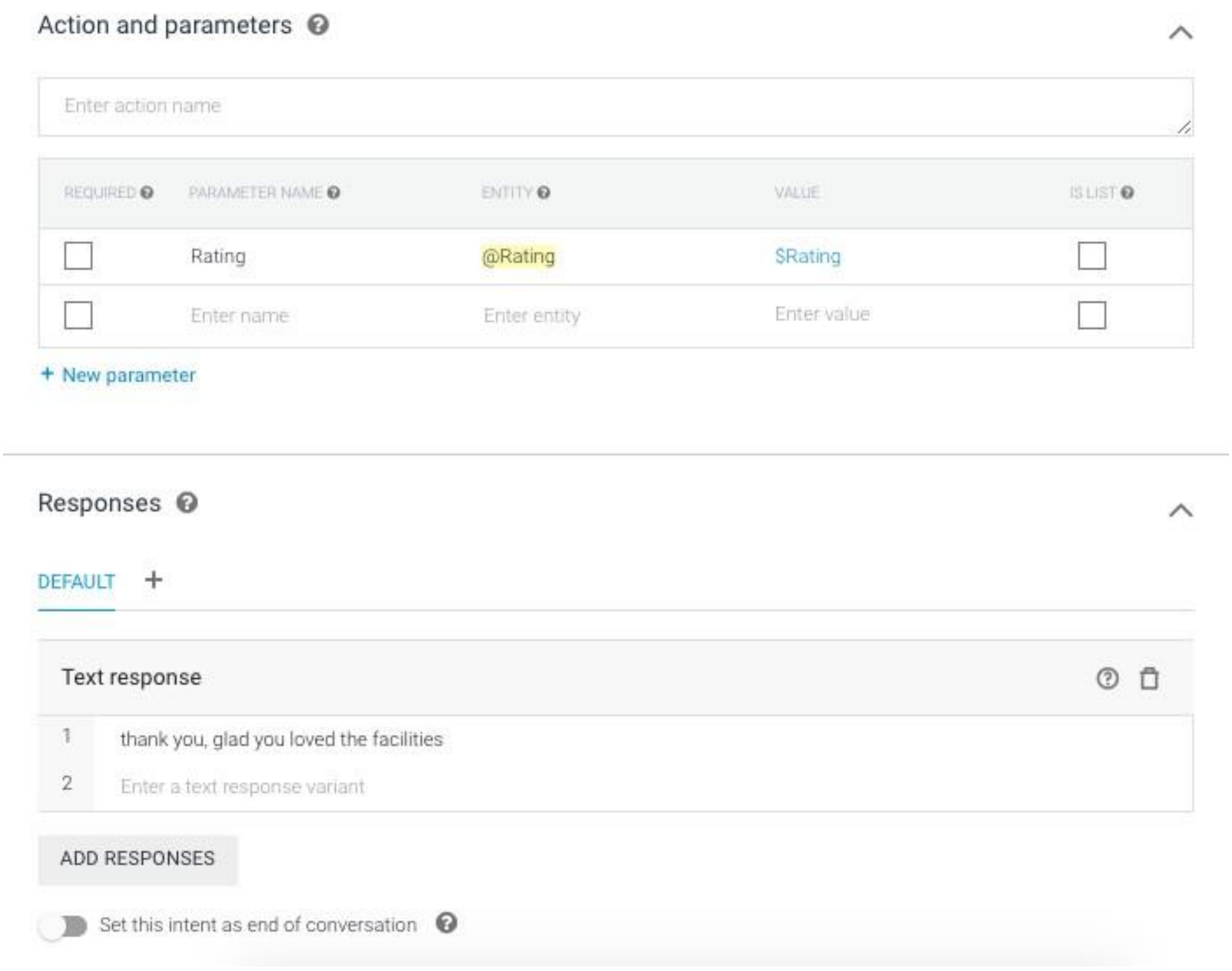
Training phrases

it was good

loved it

great

good



Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
□	Rating	@Rating	$Rating	□
□	Enter name	Enter entity	Enter value	□

+ New parameter

Responses

DEFAULT +

Text response

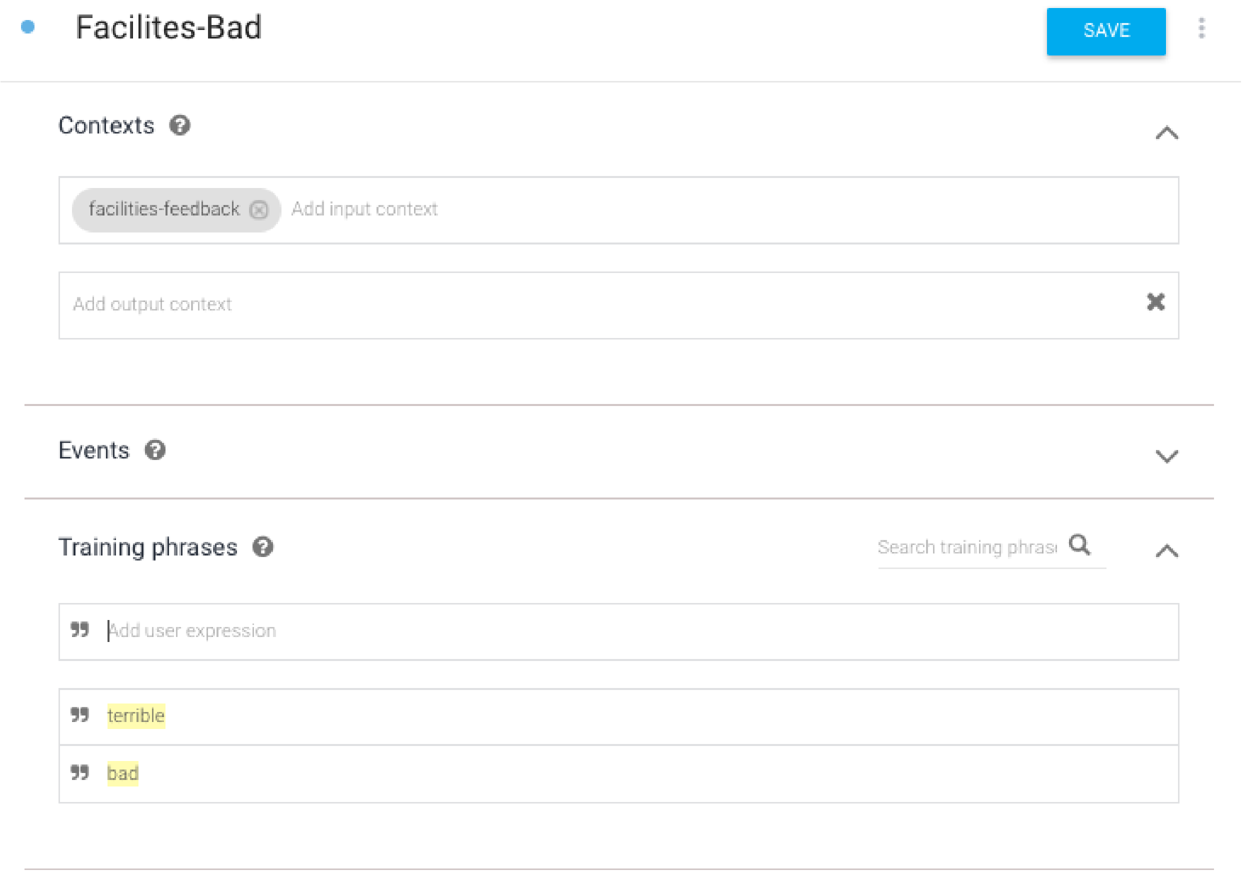
1 thank you, glad you loved the facilities

2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

New Intent : Facilities—Bad



Facilites-Bad

SAVE

Contexts

facilities-feedback ⊗ Add input context

Add output context ✖

Events

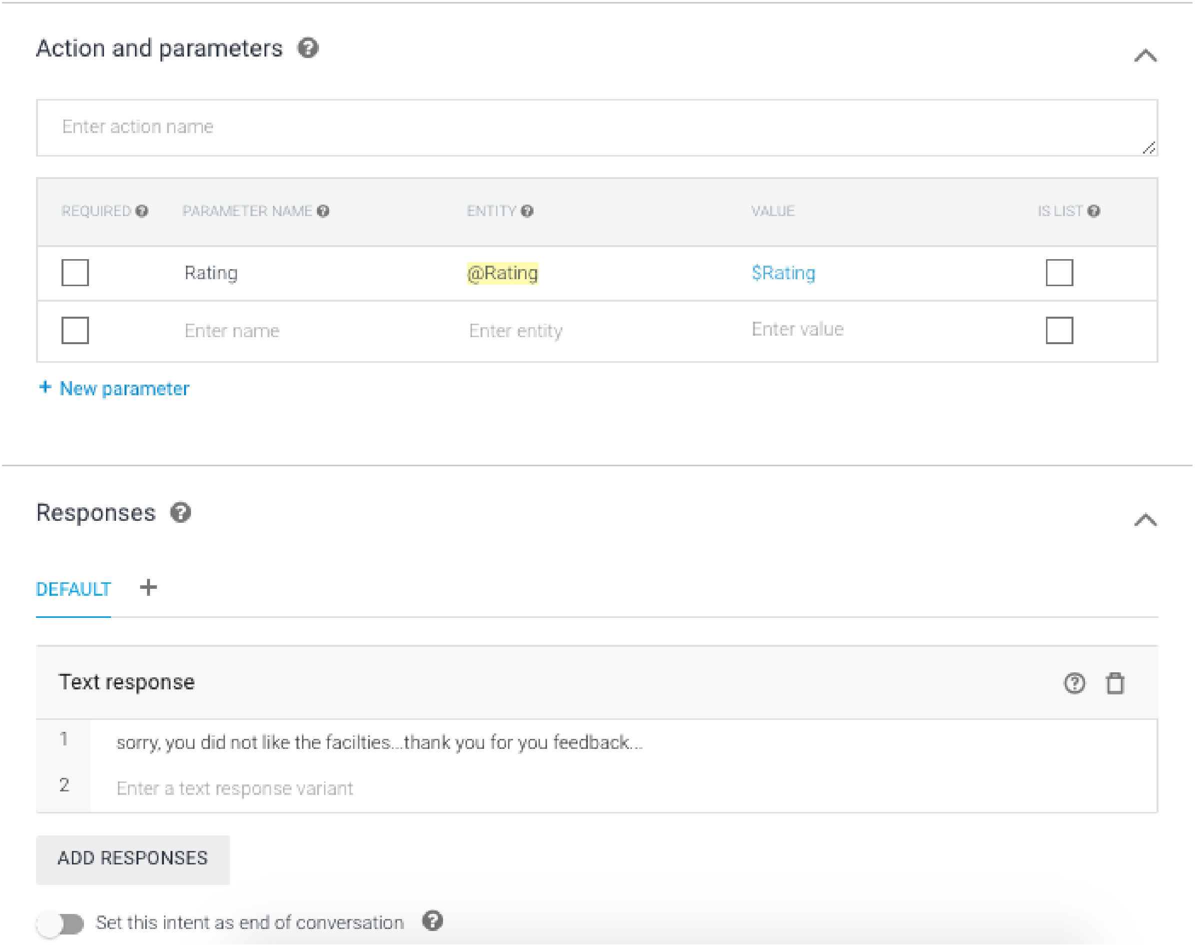
Training phrases

Search training phrase

Add user expression

terrible

bad



Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
☐	Rating	@Rating	$Rating	☐
☐	Enter name	Enter entity	Enter value	☐

+ New parameter

Responses

DEFAULT +

Text response

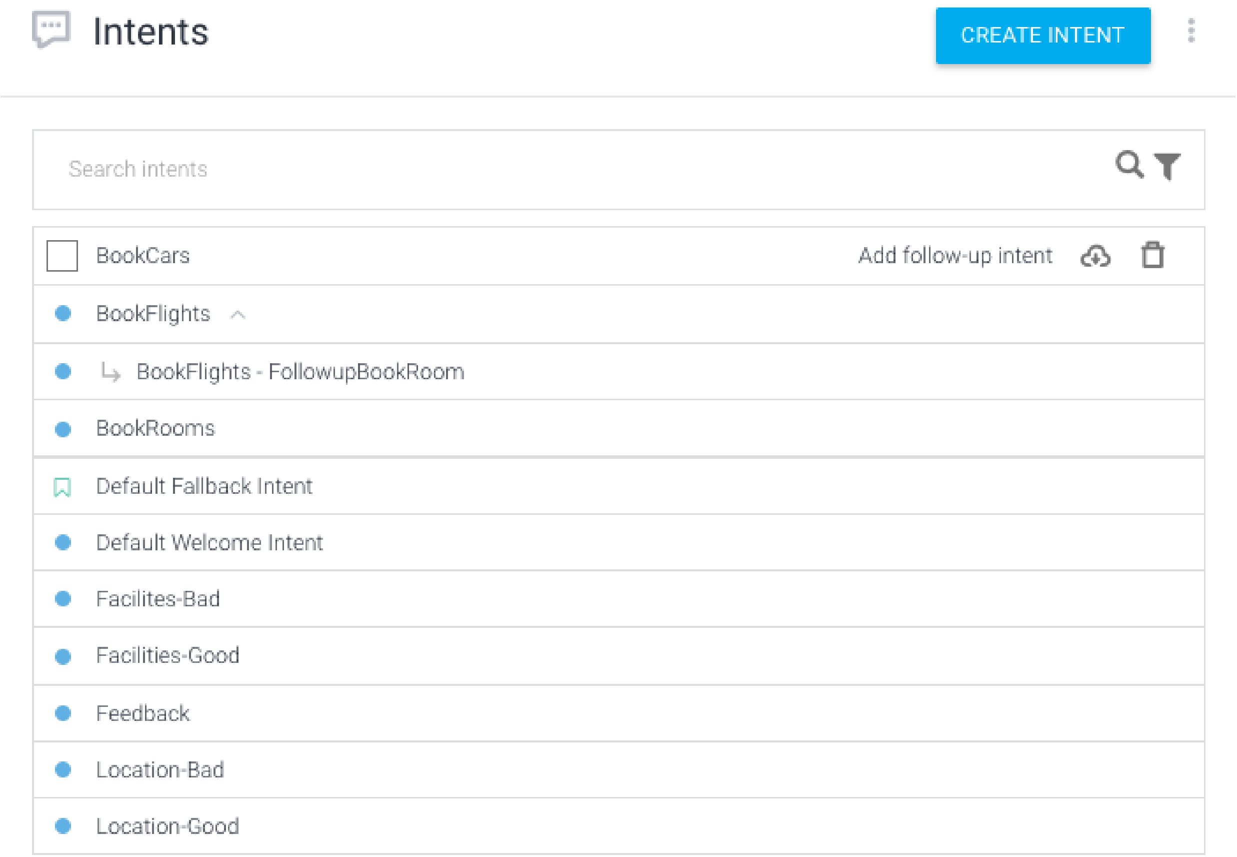
1 sorry, you did not like the facilties...thank you for you feedback...

2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation

Overall Intents:



Intents

CREATE INTENT

Search intents

- ☐ BookCars — Add follow-up intent
- ● BookFlights
- ● ↳ BookFlights - FollowupBookRoom
- ● BookRooms
- Default Fallback Intent
- ● Default Welcome Intent
- ● Facilites-Bad
- ● Facilities-Good
- ● Feedback
- ● Location-Bad
- ● Location-Good

Summary :

Dialogs can be linear or non-linear

Linear Dialogs

```
* Linear Dialogs can span single or multiple intents
* Primary reason of Linear Dialogs is to capture information from the user to
complete actions
* Context allows information sharing to simulate a more natural conversations
```

Non-Linear Dialogs

```
* Non-linear dialogs help branch to intents based on user responses
```

So, our bot can now manage different types of dialogs and might look smarter…but till now we have not connected with an external site to pul data. So in the next chapter, we shall see how we can create one more bot / Agent (StockTracker) that will connect to an external site and get us the data real-time, no more pre-defined text responses. We are entering the fulfilment world!

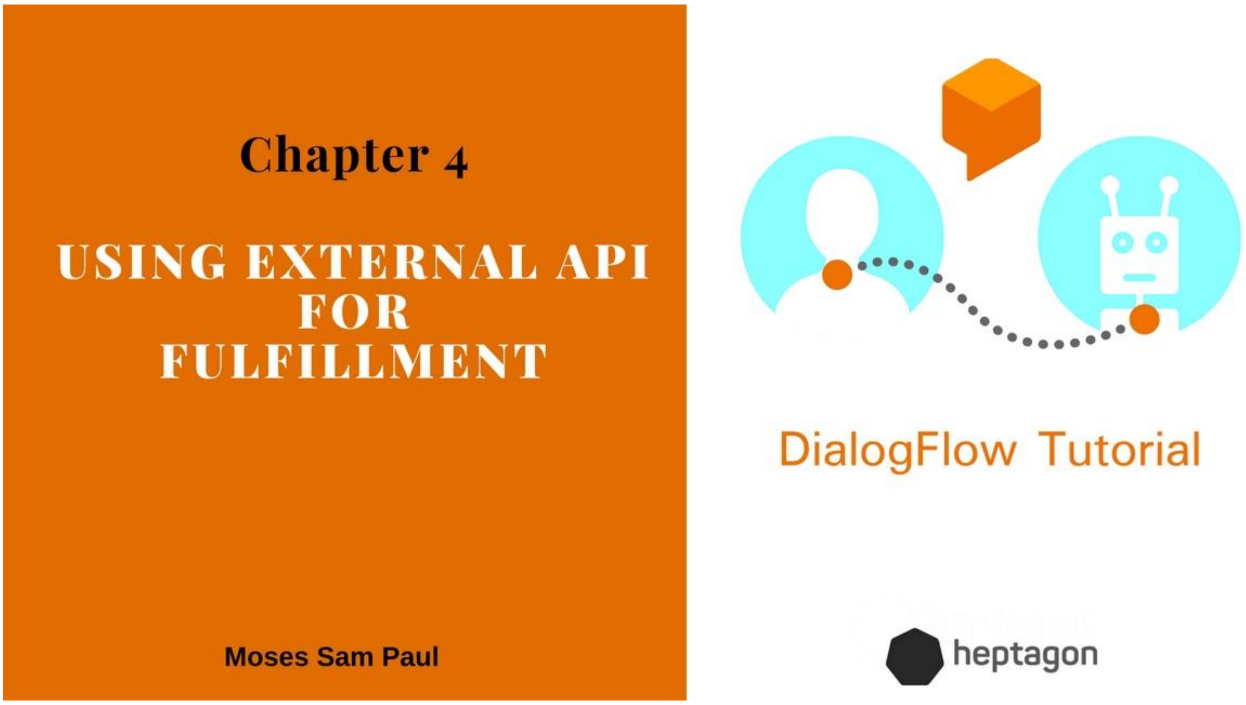
How to build a chatbot with Dialog flow | Chapter 4— External API for Fulfilment

In this chapter we are gonna create a “StockTracker” bot that will pull a specific ticker data from an external website through RESTFUl APIs. [Don’t worry if you don’t understand the jargons, we shall break it down]

Chapter 1: Basic Gyaan about Chatbots and setting up the dialog flow environment

Chapter 2: Understanding the building blocks of Dialogflow

Chapter 3: Using linear, followup and non-linear dialogs to create a “TripPlanner” bot

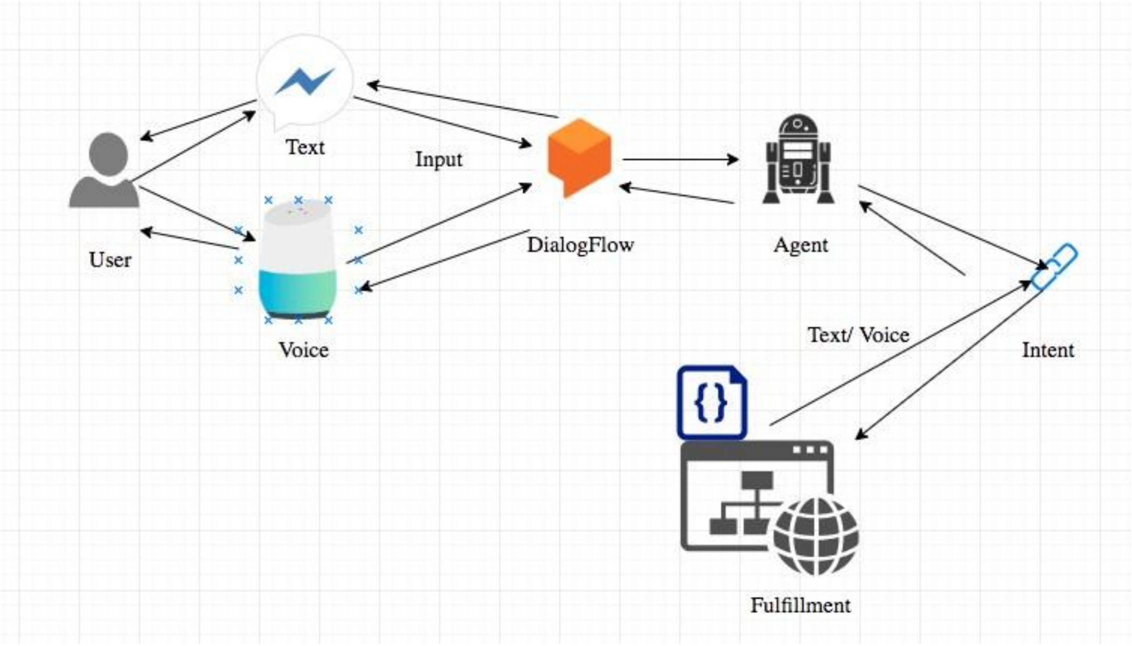


Using External API for Fulfillment

Overview of the Module:

- Completing actions requires more than a text response
- Fulfill actions using webhooks [webhooks are just HTTP callback that can be notified using post request]
- Host Code using Cloud Functions for Firebase
- Other external Endpoints such as AWS lambdas will work as well

Quick Recap:



Fulfilment

Executing the code that will complete the action requested by the user to fulfil the intent of the user.

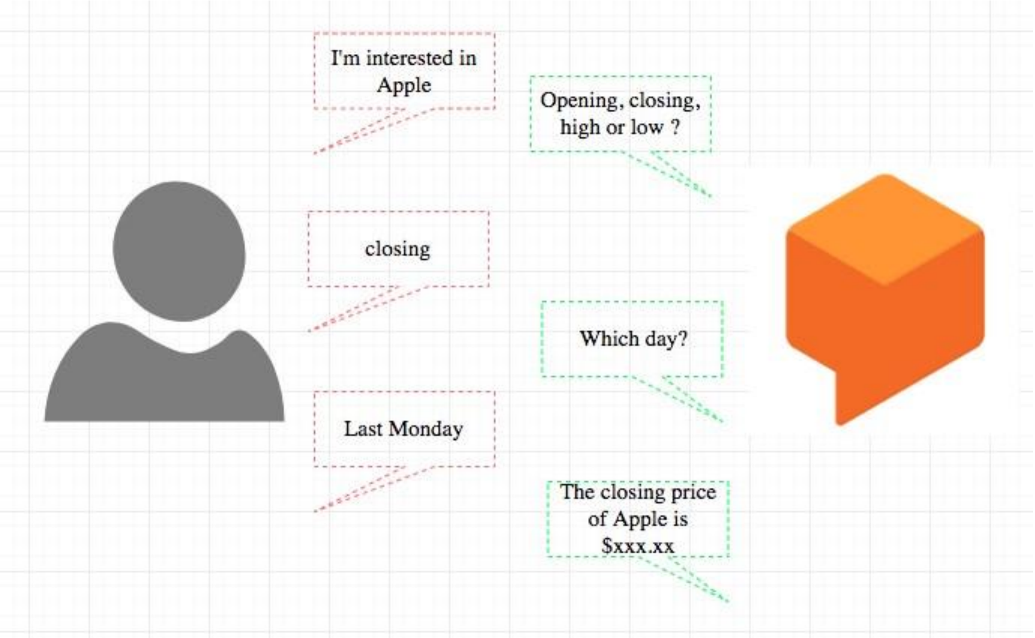
Code might require us to update the datbase

Once the code is generated a response is generated to tex response that’s sent to the user..which is in-turn sent to the agent (bot) which in turn is sent to the app that the user used to interact with your agent.

…..

Example:

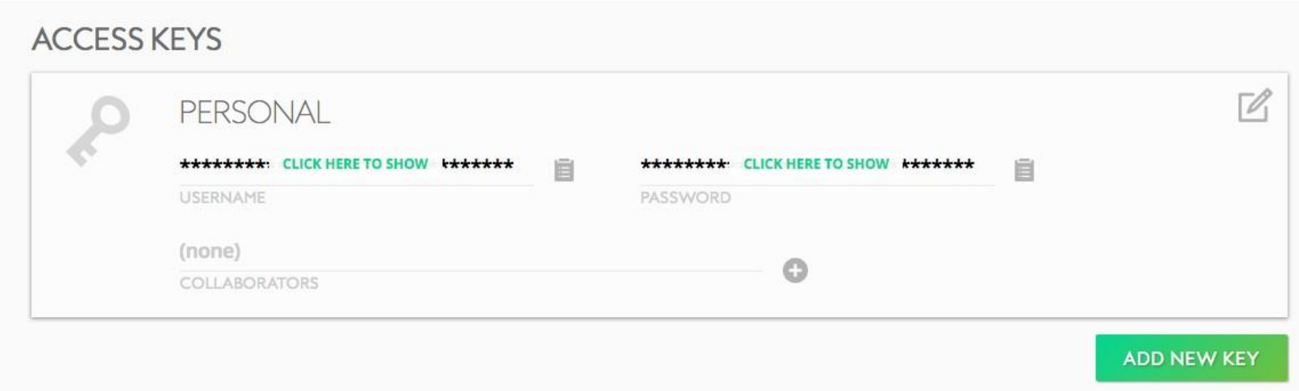
We are gonna get the stock price of Apple from an external source.


I'm interested in Apple
Opening, closing, high or low ?
closing
Which day?
Last Monday
The closing price of Apple is $xxx.xx

The Intrinio platform

Apple Inc | AAPL | Intrinio
*Apple, Inc. engages in the design, manufacture, and marketing of mobile communication, media devices, personal…*intrinio.com
Alternative for Indian Market could be: https://kite.trade/startups

Signup and go to your ccount section..you should find your API access keys



ACCESS KEYS

PERSONAL

********: CLICK HERE TO SHOW *******
USERNAME

********: CLICK HERE TO SHOW *******
PASSWORD

(none)
COLLABORATORS

API explorer:

https://intrinio.com/api-explorer?formula=historical_data&values=eyJpZGVudGlmaWVyIjoiQUFQTCIsIml0ZW0iOiIiLCJzZXF1ZW5jZSI6MH0%3D

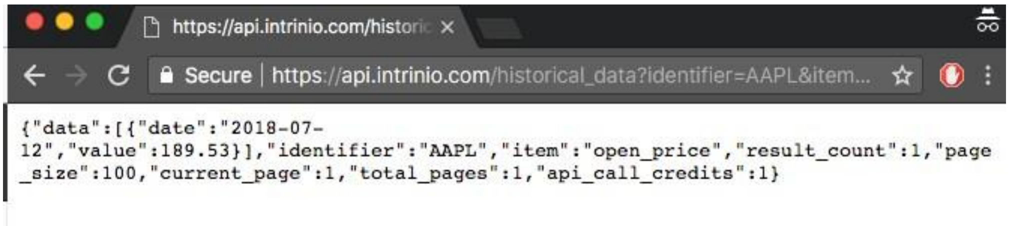
Documentation

http://docs.intrinio.com/#introduction

Test:

https://api.intrinio.com/historical_data?identifier=AAPL&item=open_price&start_date=2018-07-12&end_date=2018-07-12

A prompt will ask for your username and password …get it from your "my account" page and you will get aJSON response as below.



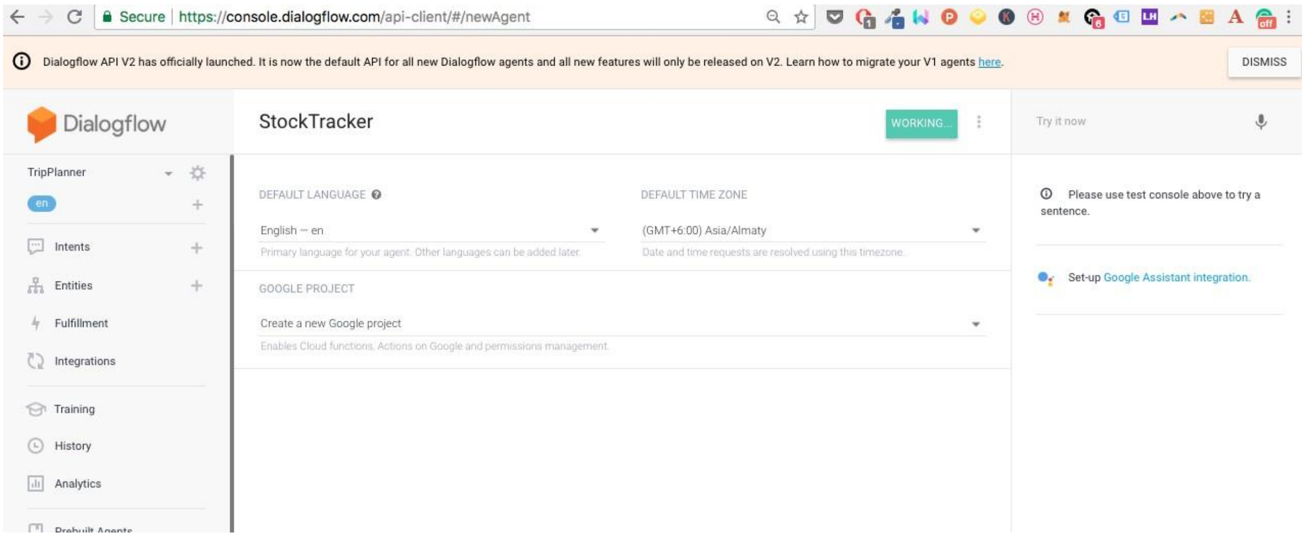
https://api.intrinio.com/historic
Secure | https://api.intrinio.com/historical_data?identifier=AAPL&item...

```
{"data":[{"date":"2018-07-12","value":189.53}],"identifier":"AAPL","item":"open_price","result_count":1,"page_size":100,"current_page":1,"total_pages":1,"api_call_credits":1}
```

Demo

- Setup fulfilment for the stockmarket tracker using cloud functions for firebase.
- Use the Blaze pay as you go tier to access external APIs

Step 1: Setup a New Agent in Dialog Flow



Secure | https://console.dialogflow.com/api-client/#/newAgent

Dialogflow API V2 has officially launched. It is now the default API for all new Dialogflow agents and all new features will only be released on V2. Learn how to migrate your V1 agents here.

DEFAULT LANGUAGE: English — en
Primary language for your agent. Other languages can be added later.

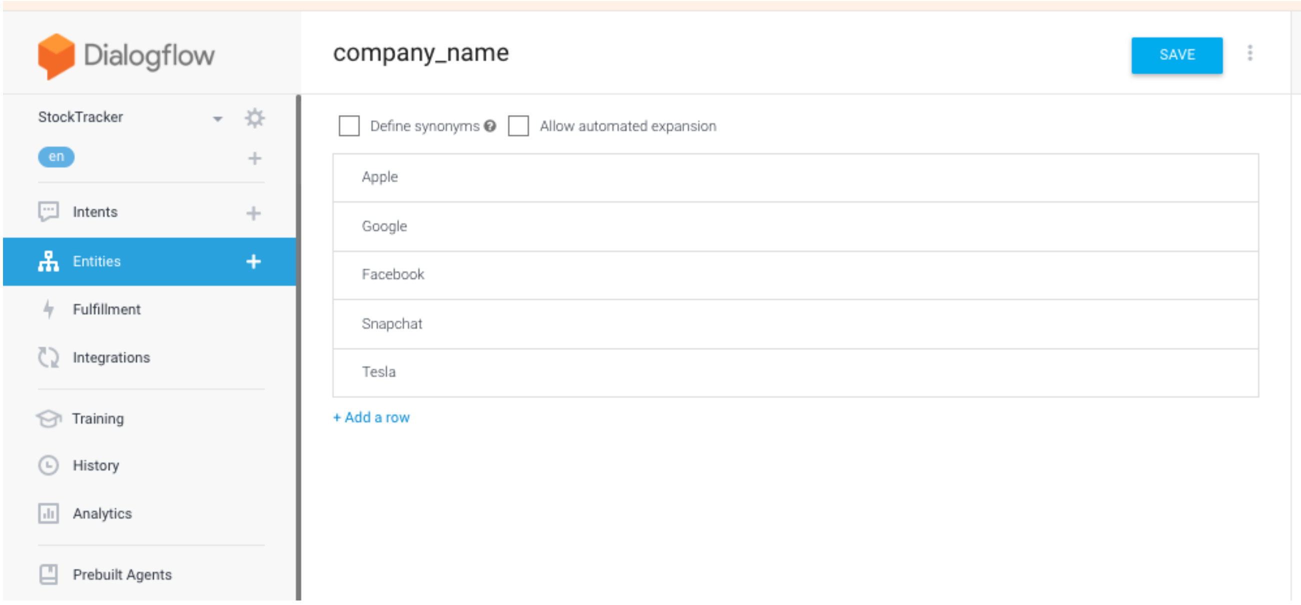
DEFAULT TIME ZONE: (GMT+6:00) Asia/Almaty
Date and time requests are resolved using this timezone.

GOOGLE PROJECT: Create a new Google project
Enables Cloud functions, Actions on Google and permissions management.

Please use test console above to try a sentence.

Set-up Google Assistant integration.

Step 2: Setup entities

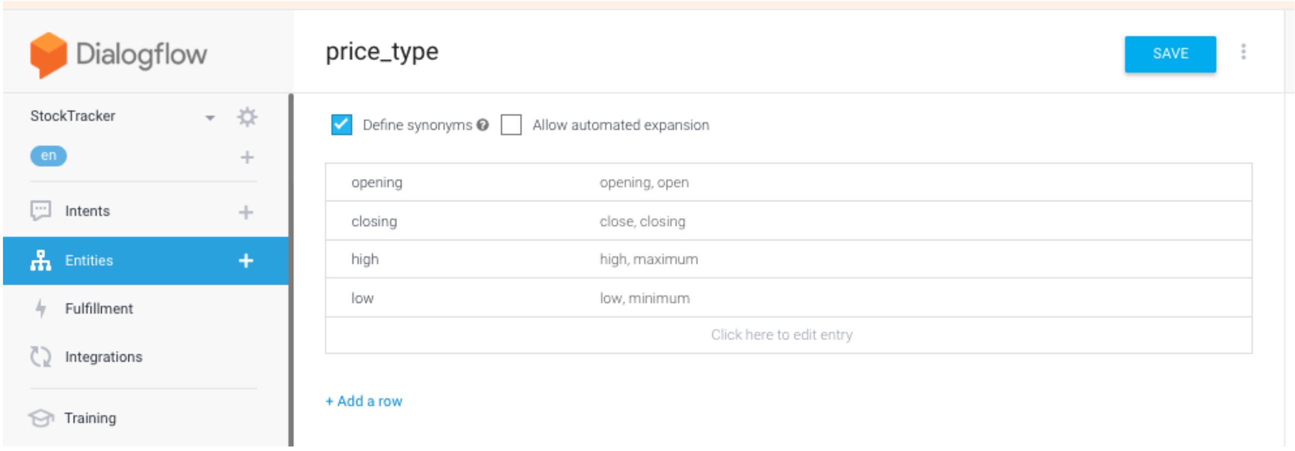


company_name

□ Define synonyms □ Allow automated expansion

Apple
Google
Facebook
Snapchat
Tesla

+ Add a row



price_type

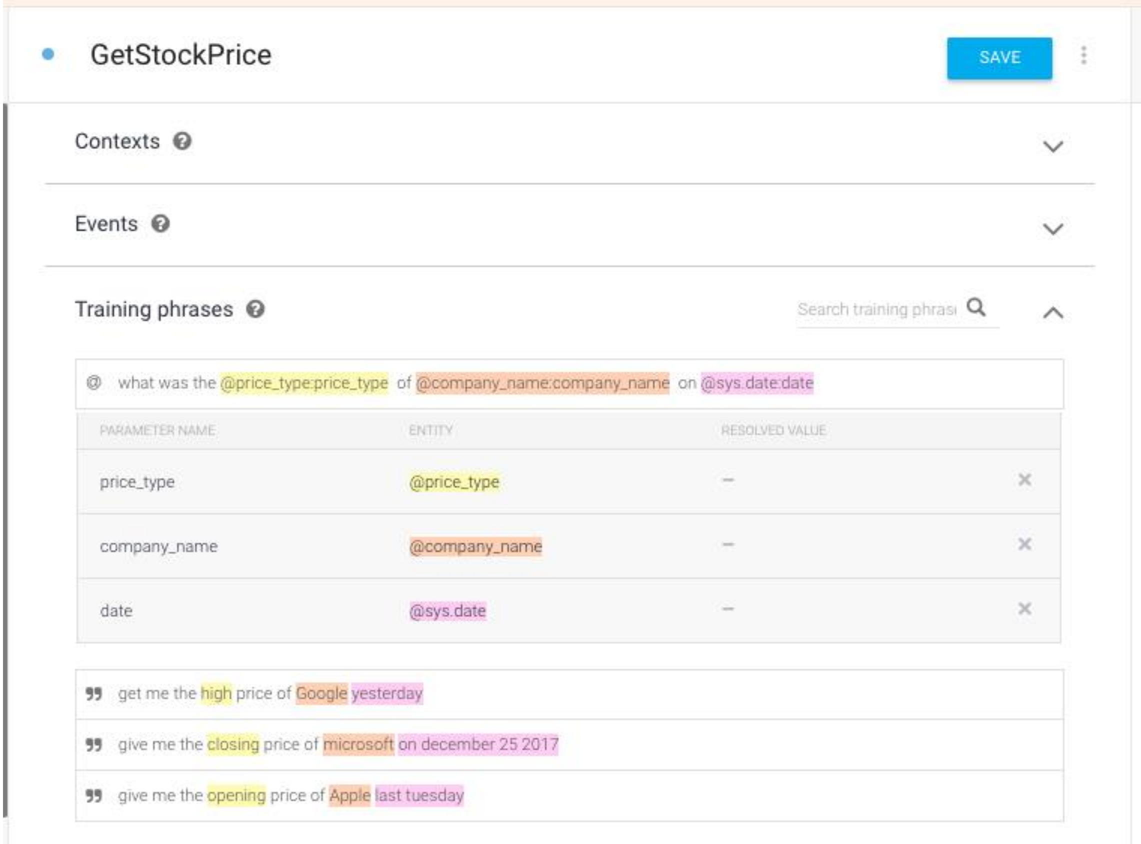
☑ Define synonyms □ Allow automated expansion

opening	opening, open
closing	close, closing
high	high, maximum
low	low, minimum

Click here to edit entry

+ Add a row

Step 3: Intent



GetStockPrice

Contexts

Events

Training phrases

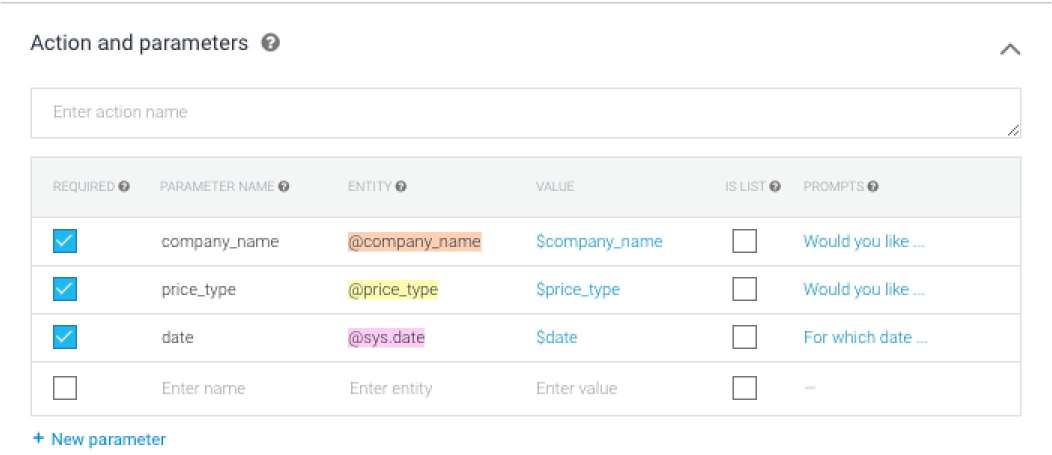
what was the @price_type:price_type of @company_name:company_name on @sys.date:date

PARAMETER NAME	ENTITY	RESOLVED VALUE
price_type	@price_type	–
company_name	@company_name	–
date	@sys.date	–

get me the high price of Google yesterday

give me the closing price of microsoft on december 25 2017

give me the opening price of Apple last tuesday

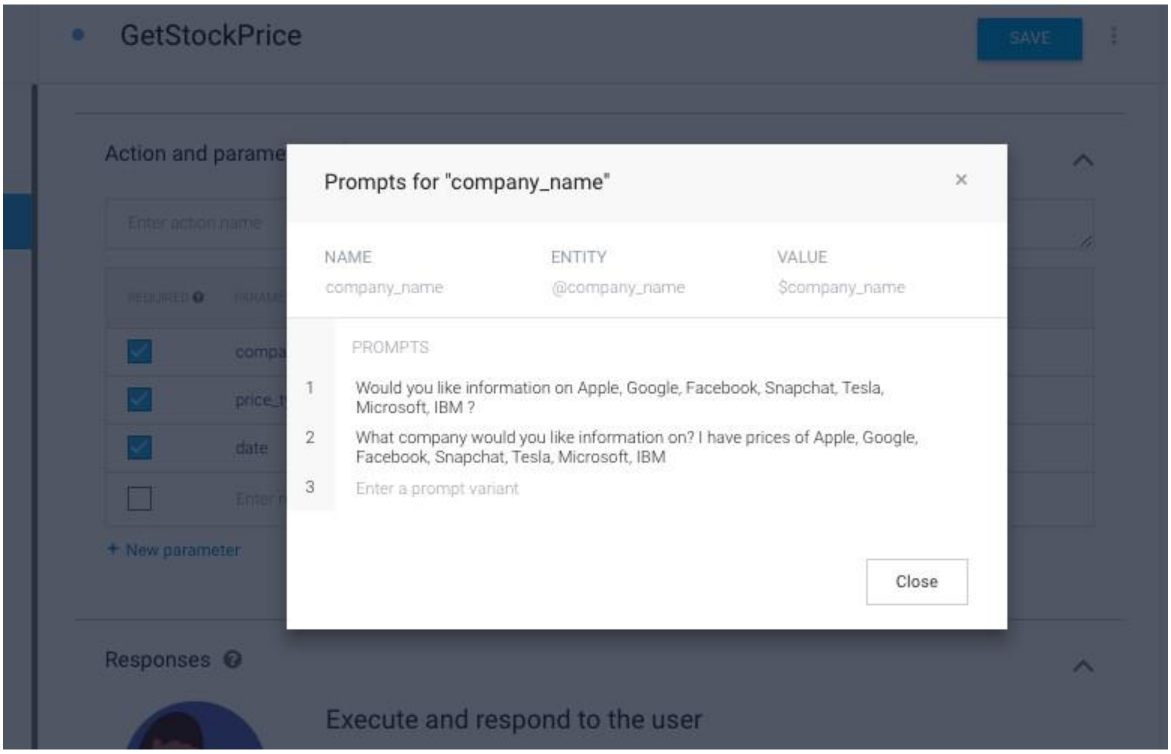


Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	company_name	@company_name	$company_name	□	Would you like ...
☑	price_type	@price_type	$price_type	□	Would you like ...
☑	date	@sys.date	$date	□	For which date ...
□	Enter name	Enter entity	Enter value	□	—

+ New parameter



Prompts for "company_name"

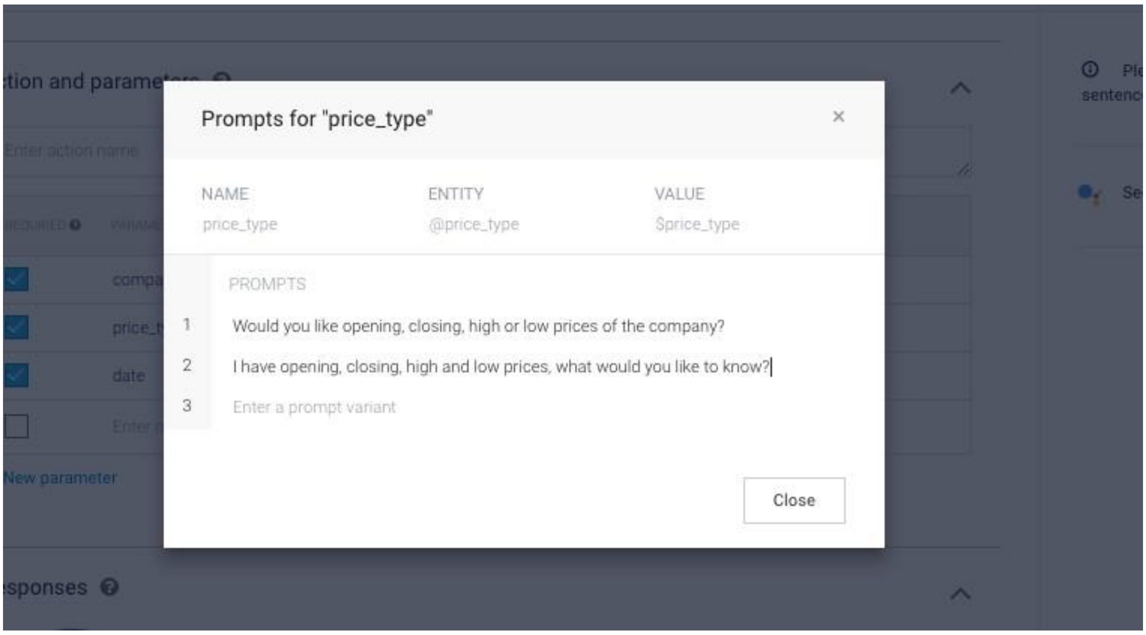
NAME	ENTITY	VALUE
company_name	@company_name	$company_name

PROMPTS

1 Would you like information on Apple, Google, Facebook, Snapchat, Tesla, Microsoft, IBM ?

2 What company would you like information on? I have prices of Apple, Google, Facebook, Snapchat, Tesla, Microsoft, IBM

3 Enter a prompt variant



Prompts for "price_type"

NAME	ENTITY	VALUE
price_type	@price_type	$price_type

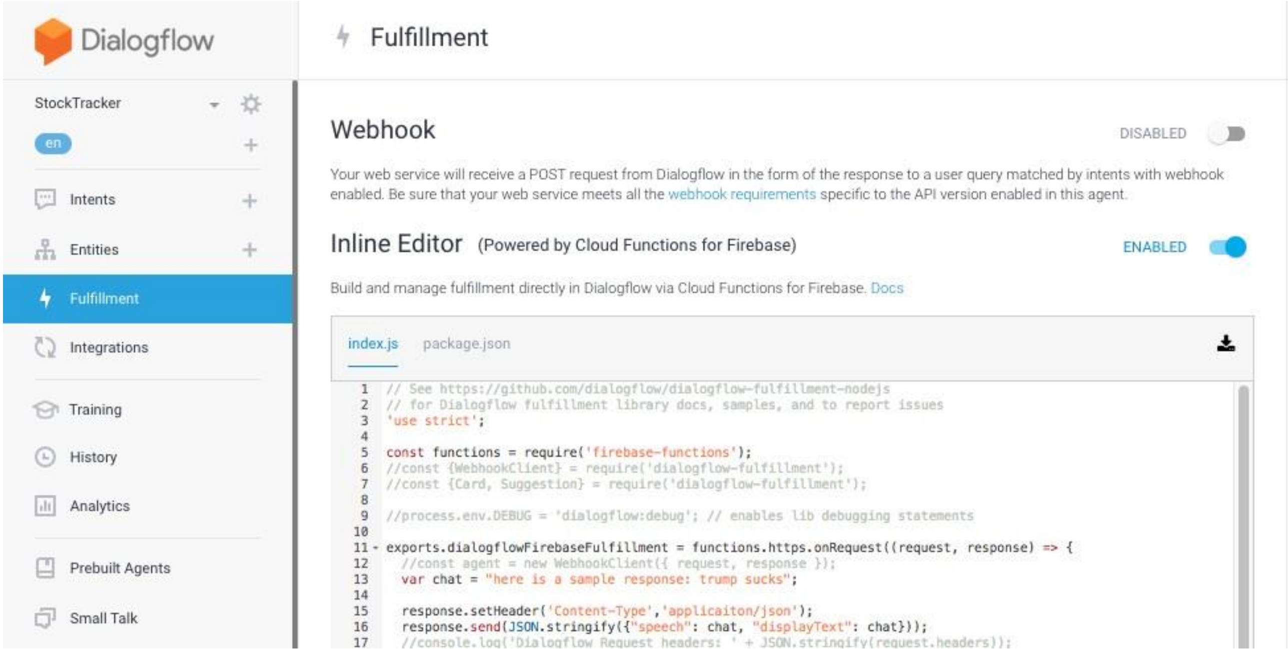
PROMPTS

1 Would you like opening, closing, high or low prices of the company?

2 I have opening, closing, high and low prices, what would you like to know?

3 Enter a prompt variant

Fulfillment:



Webhook

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.

Inline Editor (Powered by Cloud Functions for Firebase)

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. Docs

```
// See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
// for Dialogflow fulfillment library docs, samples, and to report issues
'use strict';

const functions = require('firebase-functions');
//const {WebhookClient} = require('dialogflow-fulfillment');
//const {Card, Suggestion} = require('dialogflow-fulfillment');

//process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements

exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
  //const agent = new WebhookClient({ request, response });
  var chat = "here is a sample response: trump sucks";

  response.setHeader('Content-Type','applicaiton/json');
  response.send(JSON.stringify({"speech": chat, "displayText": chat}));
  //console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
```

```
'use strict';

const functions = require('firebase-functions');
//const {WebhookClient} = require('dialogflow-fulfillment');
//const {Card, Suggestion} = require('dialogflow-fulfillment');

//process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements

exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
  //const agent = new WebhookClient({ request, response });
  var chat = "here is a sample response: trump sucks";

response.setHeader('Content-Type','applicaiton/json');
  response.send(JSON.stringify({"fulfillmentText": chat}));
});
```

<Error Log: If you are following a dialogflow v1 tutorial you might be using response.send(JSON.stringify({“speech”: chat, “displayText”, chat}));

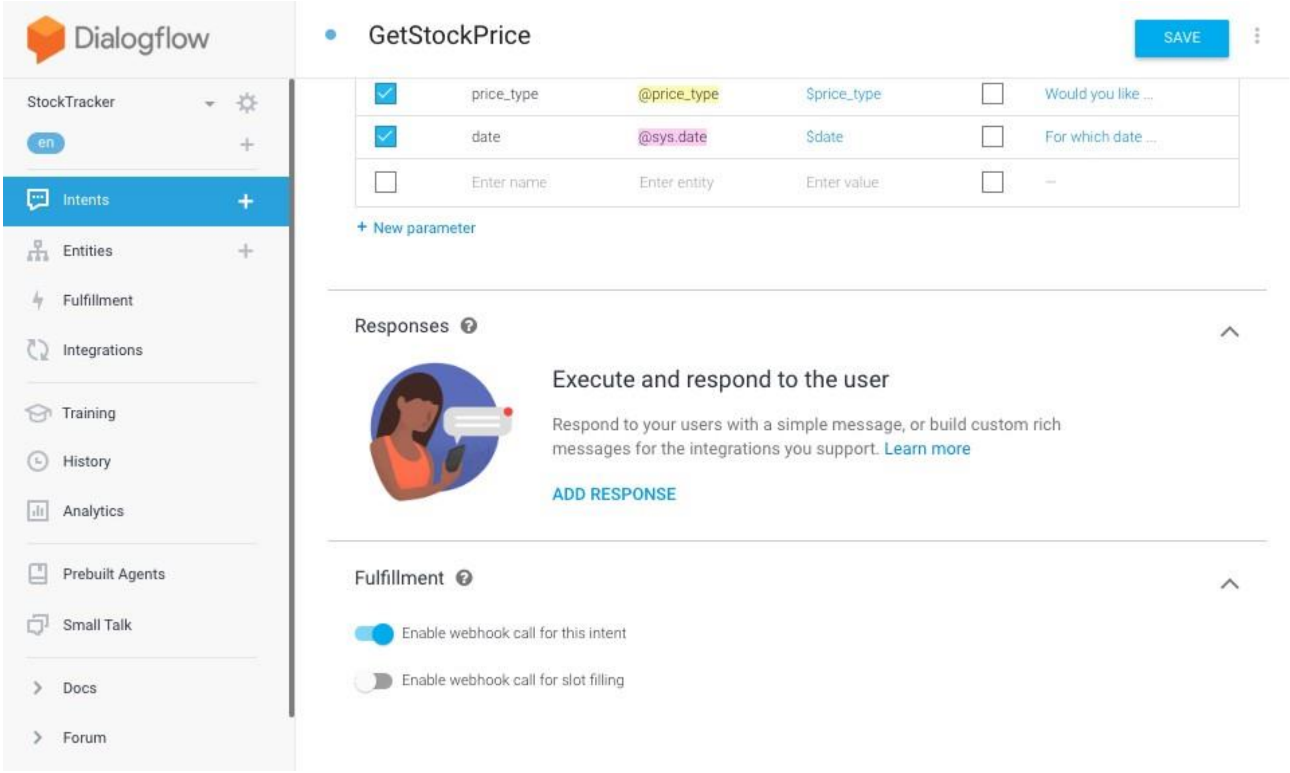
That will throw an error

```
Webhook call failed. Error: Failed to parse webhook JSON response: Cannot find field: speech in message
google.cloud.dialogflow.v2.WebhookResponse.
```

That’s because speech and displaytext has been replaced with “fulfillmentText” in v2 of Dialogflow.

>

Enable webhook as fulfillment within the GetStockPrice intent



GetStockPrice

☑	price_type	@price_type	$price_type	□	Would you like ...
☑	date	@sys.date	$date	□	For which date ...
□	Enter name	Enter entity	Enter value	□	—

+ New parameter

Responses

Execute and respond to the user

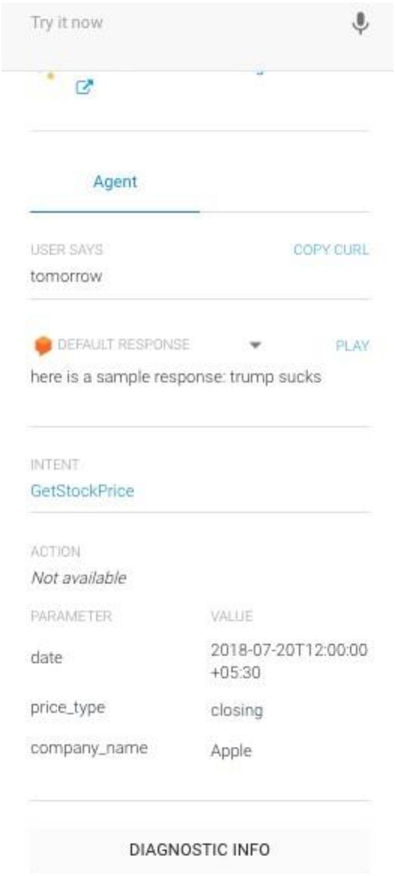
Respond to your users with a simple message, or build custom rich messages for the integrations you support. Learn more

ADD RESPONSE

Fulfillment

● Enable webhook call for this intent

○ Enable webhook call for slot filling



USER SAYS
tomorrow

DEFAULT RESPONSE
here is a sample response: trump sucks

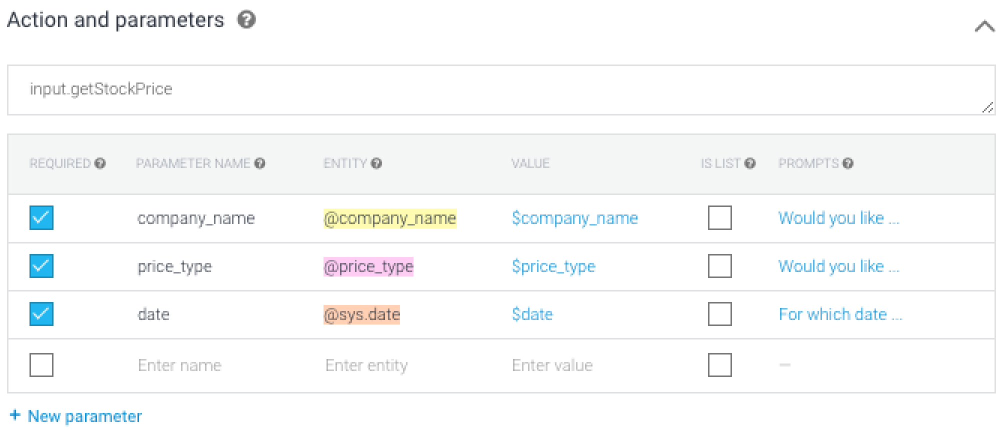
INTENT
GetStockPrice

ACTION
Not available

PARAMETER	VALUE
date	2018-07-20T12:00:00 +05:30
price_type	closing
company_name	Apple

Now we gotta make sure a specific action is named and that is called during the webhook.

Set the action within the intent’s action and parameters section: <input.getStockPrice>



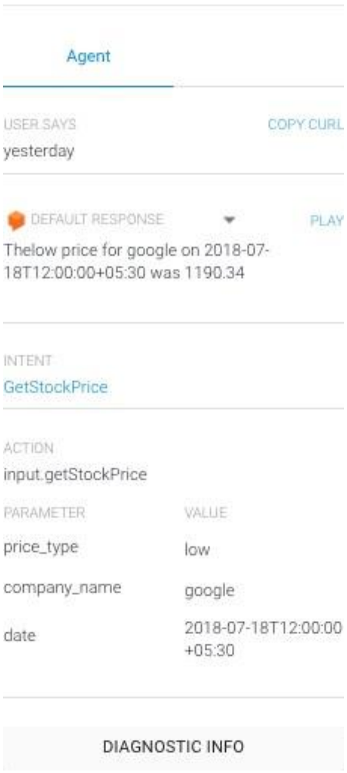
Action and parameters

input.getStockPrice

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
☑	company_name	@company_name	$company_name	□	Would you like ...
☑	price_type	@price_type	$price_type	□	Would you like ...
☑	date	@sys.date	$date	□	For which date ...
□	Enter name	Enter entity	Enter value	□	–

Code:

Output:



USER SAYS
yesterday

DEFAULT RESPONSE
Thelow price for google on 2018-07-18T12:00:00+05:30 was 1190.34

INTENT
GetStockPrice

ACTION
input.getStockPrice

PARAMETER	VALUE
price_type	low
company_name	google
date	2018-07-18T12:00:00 +05:30

Few Errors you might come across:

1. Upgrade to Blaze Pay as you go

```
Error:
Billing account not configured. External network is not accessible and quotas
are severely limited. Configure billing account to remove these restrictions
```

2. Cannot read property action

Had initially written

```
let action = request.body.result.action;
```

and got this error:

```
TypeError: Cannot read property 'action' of undefined
    at exports.dialogflowFirebaseFulfillment.functions.https.onRequest
(/user_code/index.js:9:37)
    at cloudFunction (/user_code/node_modules/firebase-
functions/lib/providers/https.js:26:47)
    at /var/tmp/worker/worker.js:684:7
    at /var/tmp/worker/worker.js:668:9
    at _combinedTickCallback (internal/process/next_tick.js:73:7)
    at process._tickDomainCallback (internal/process/next_tick.js:128:9)
```

but printing the header and the body of requests showed that dialogflow v2 has changed things a little bit.

```
Request  Body:  {"responseId":"b539a7bf-eccc-43a9-9f6b-a1fe6ea41763","queryResult":
{"queryText":"yesterday","action":"input.getStockPrice","parameters":
{"date":"2018-07-
18T12:00:00+05:30","company_name":"apple","price_type":"high"},"allRequiredParamsPresent":true,"fulfillmentMessages":
[{"text":{"text":[""]}}],"intent":{"name":"projects/stocktracker-9268f/agent/intents/7ccfc0ff-7271-4a60-83ce-
868d6eea03a0","displayName":"GetStockPrice"},"intentDetectionConfidence":1,"languageCode":"en"},"originalDetectIntentRequest":
{"payload":{}},"session":"projects/stocktracker-9268f/agent/sessions/4c4870db-0d23-ecc0-823e-de2bde1716f9"}
```

so instead of “query” add “queryResult”

```
let action = request.body.Queryresult.action;
```

Summary:

So what did we learn in this chapter? We used our bot to parse the user request and used dialogflow’s fulfilment module to create a cloud function hosted on firebase to pull data from a third party site through a RESTFUL API and revert to the user.

Wait, but did we really though ? Till now we haven’t distributed / deployed out bot in an external environment be it on slack or twitter to let the user really interact with our bot.

In the next chapter, we will integrate / deploy / distribute our app on a slack channel and let users interact with our chatbot.

How to build a chatbot with Dialog flow | Chapter 5— Connect Chatbot with Slack

July 25, 2018

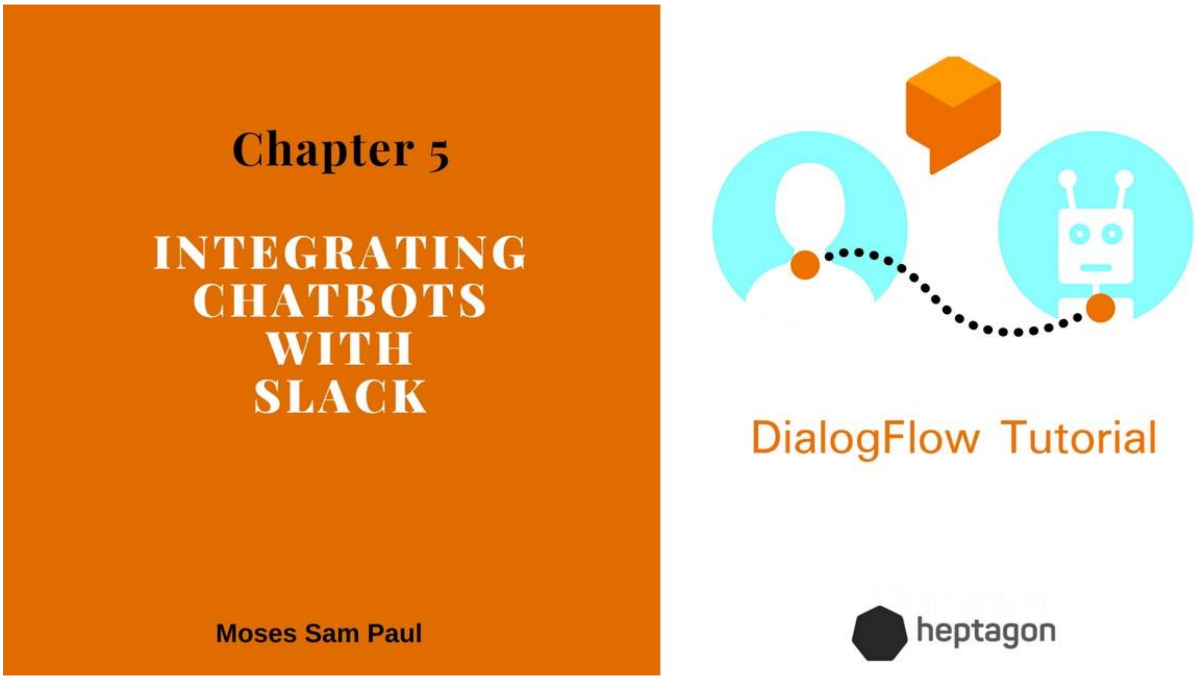
In this chapter, we are going to deploy our “stocktracker” bot to a slack channel so that external users can actually interact with our bot.

Chapter 1: Basic Gyaan about Chatbots and setting up the dialog flow environment

Chapter 2: Understanding the building blocks of Dialogflow

Chapter 3: Using linear, followup and non-linear dialogs to create a “TripPlanner” bot

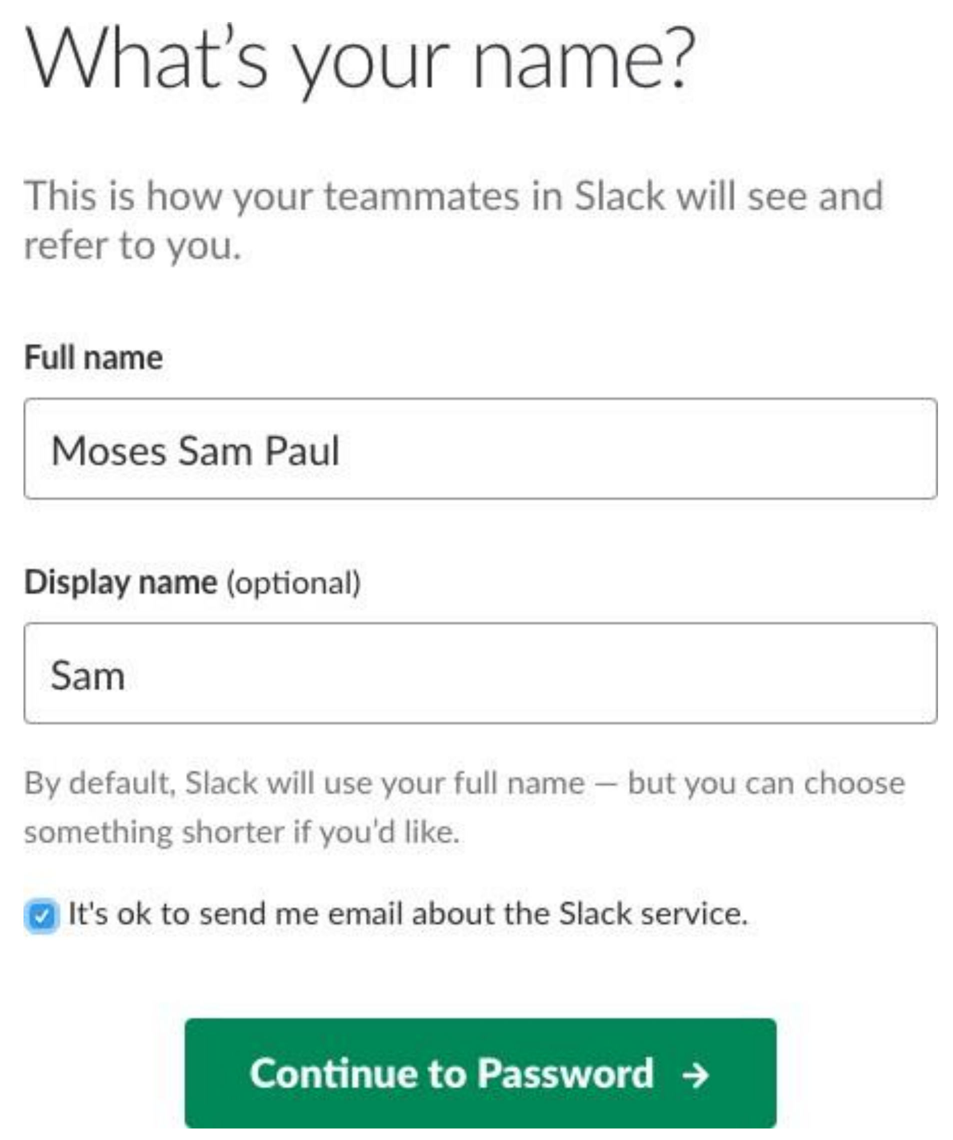
Chapter 4: We created a new bot “StockTracker” bot to pull data from an external site through fulfilment and revert the result to the user



Integrating with third-party Applications

Slack, Kik, Viber, Skype, Google Assistant.

1. Slack workspace



What's your name?

This is how your teammates in Slack will see and refer to you.

Full name

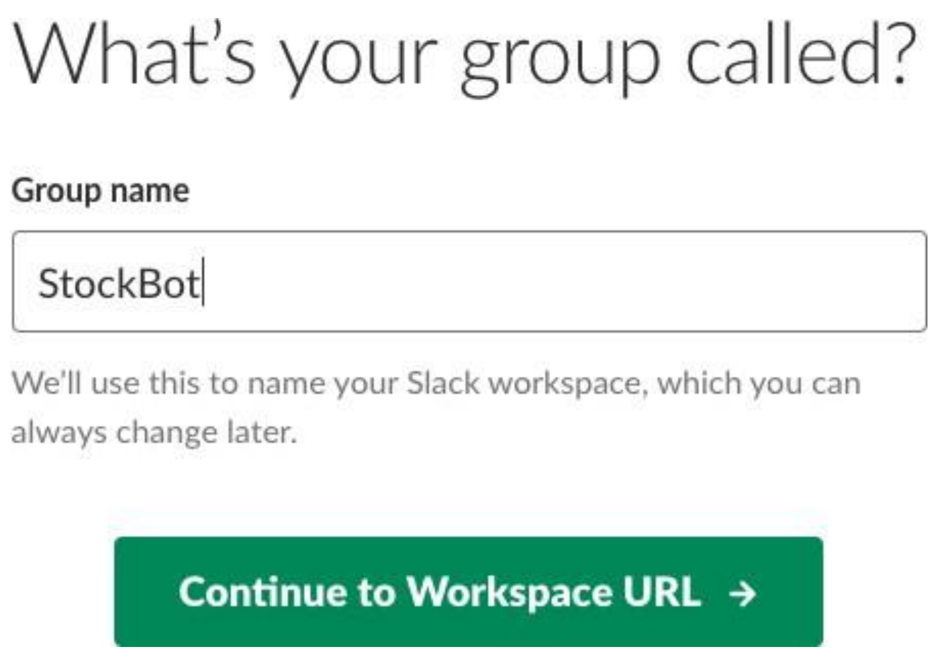
Moses Sam Paul

Display name (optional)

Sam

By default, Slack will use your full name — but you can choose something shorter if you'd like.

☑ It's ok to send me email about the Slack service.

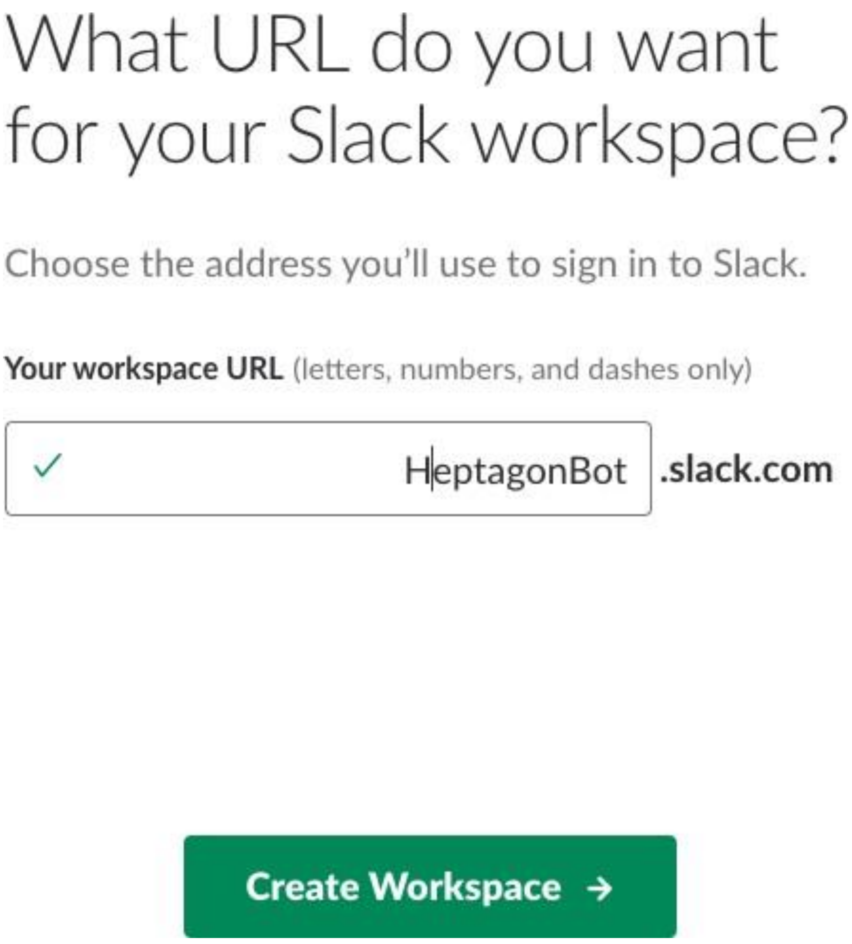


What's your group called?

Group name

StockBot

We'll use this to name your Slack workspace, which you can always change later.



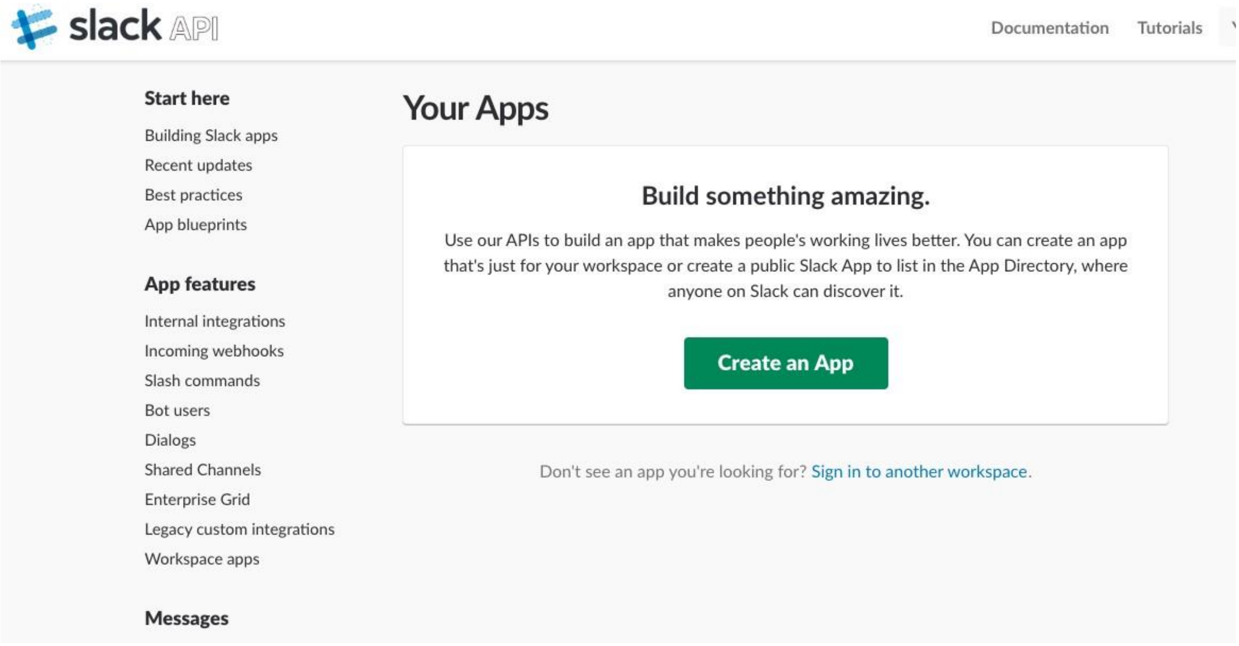
What URL do you want for your Slack workspace?

Choose the address you'll use to sign in to Slack.

Your workspace URL (letters, numbers, and dashes only)

✓ HeptagonBot .slack.com

2. Create and app on slack—https://api.slack.com

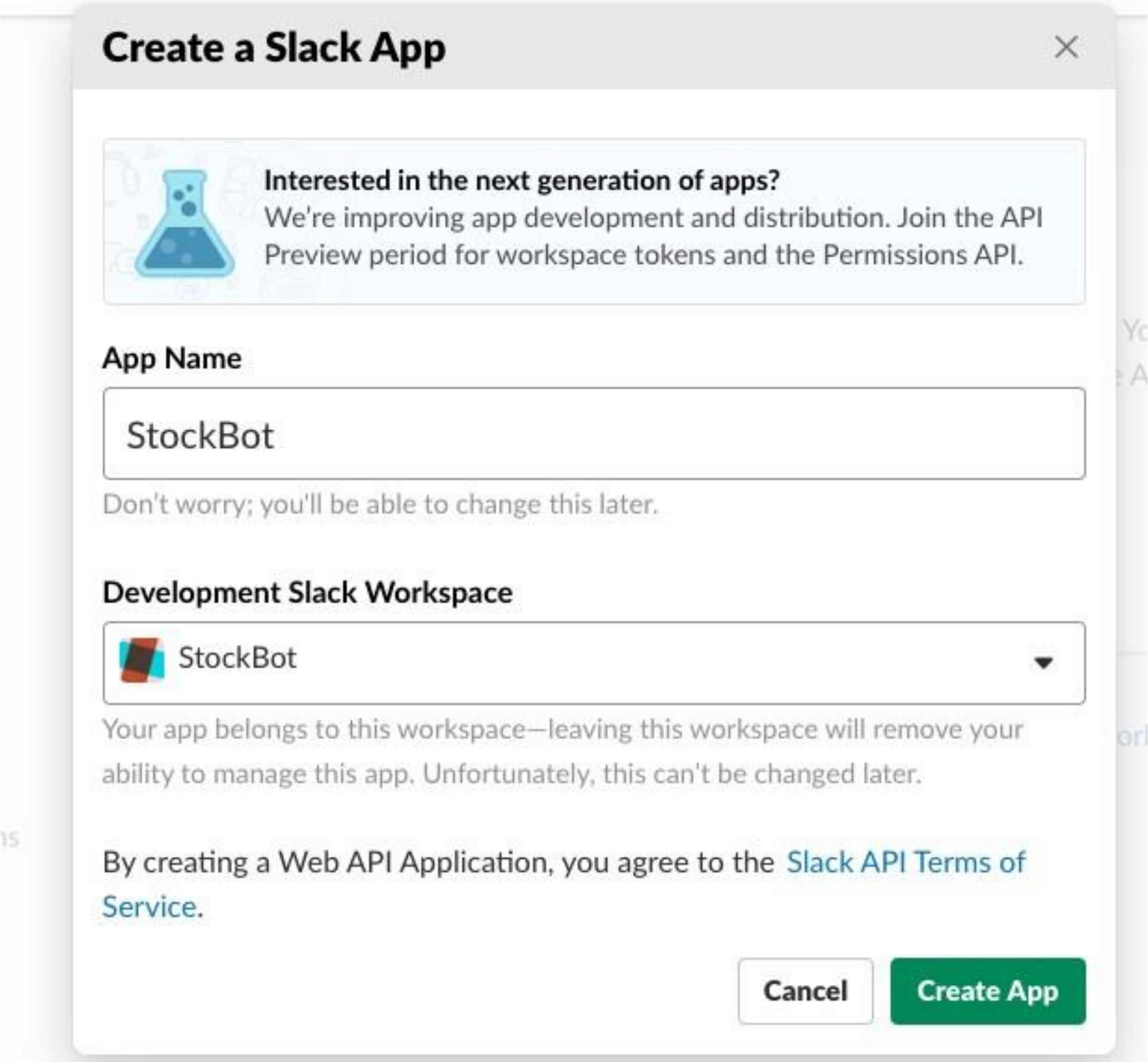


Your Apps

Build something amazing.

Use our APIs to build an app that makes people's working lives better. You can create an app that's just for your workspace or create a public Slack App to list in the App Directory, where anyone on Slack can discover it.

Don't see an app you're looking for? Sign in to another workspace.



Create a Slack App

Interested in the next generation of apps?
We're improving app development and distribution. Join the API Preview period for workspace tokens and the Permissions API.

App Name

StockBot

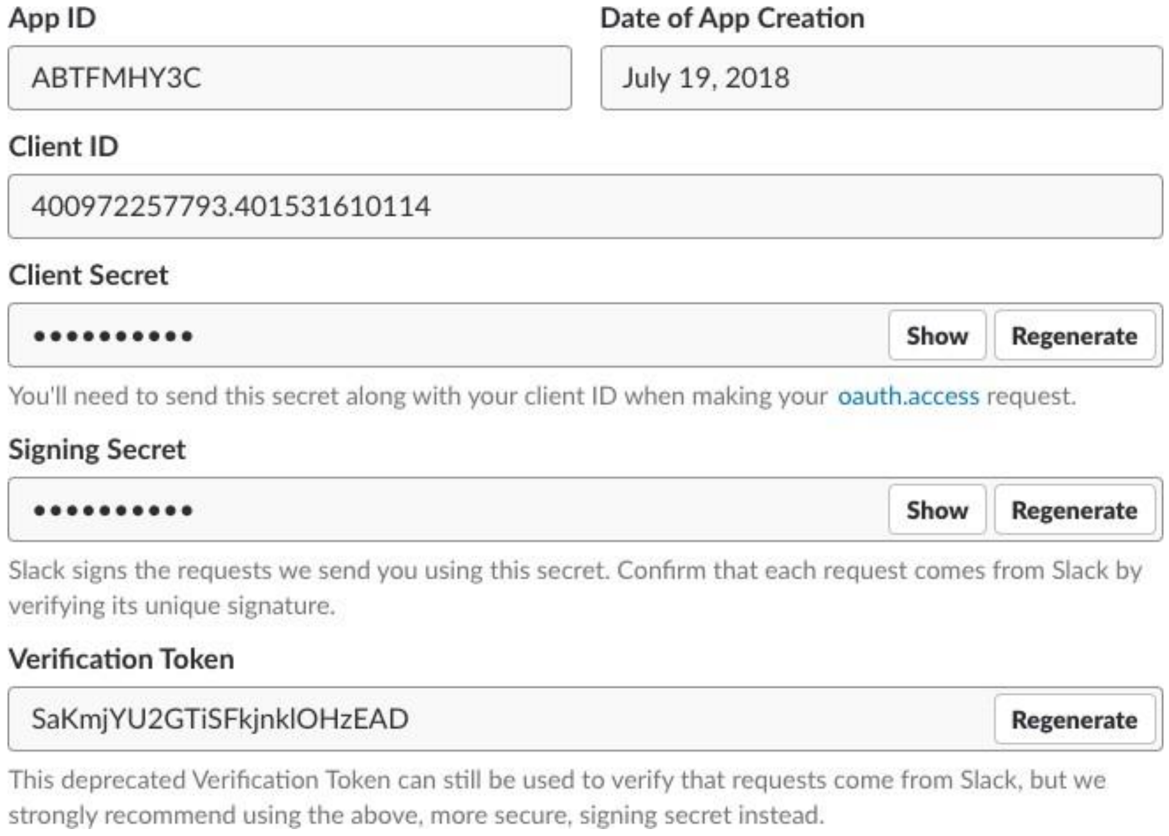
Don't worry; you'll be able to change this later.

Development Slack Workspace

StockBot

Your app belongs to this workspace—leaving this workspace will remove your ability to manage this app. Unfortunately, this can't be changed later.

By creating a Web API Application, you agree to the Slack API Terms of Service.



App ID

ABTFMHY3C

Date of App Creation

July 19, 2018

Client ID

400972257793.401531610114

Client Secret

••••••••••

You'll need to send this secret along with your client ID when making your oauth.access request.

Signing Secret

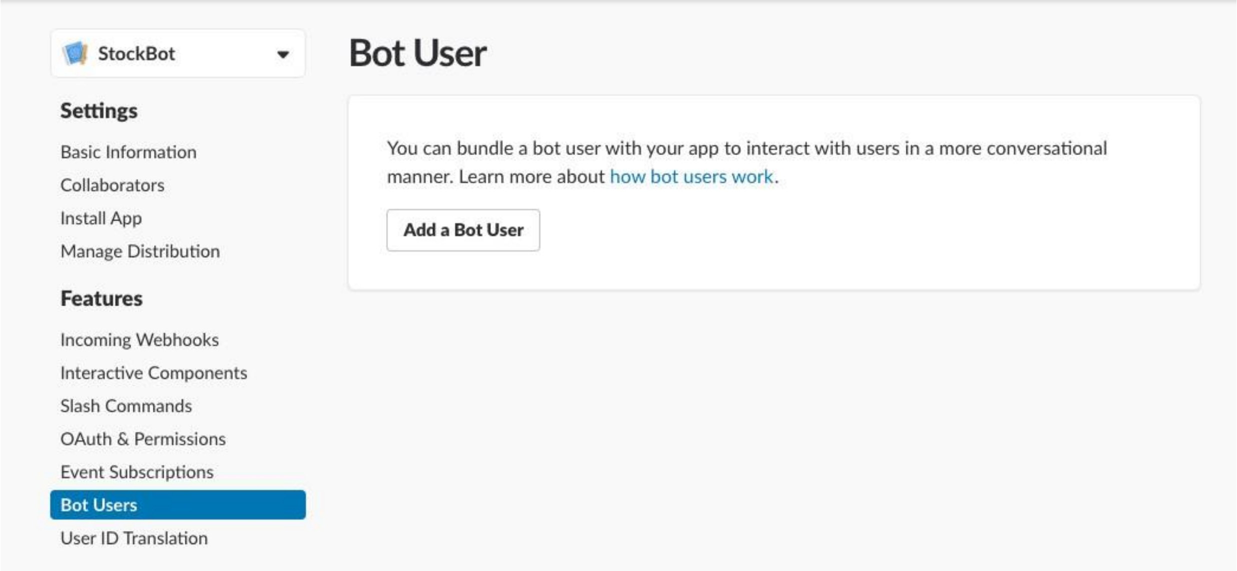
••••••••••

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token

SaKmjYU2GTiSFkjnklOHzEAD

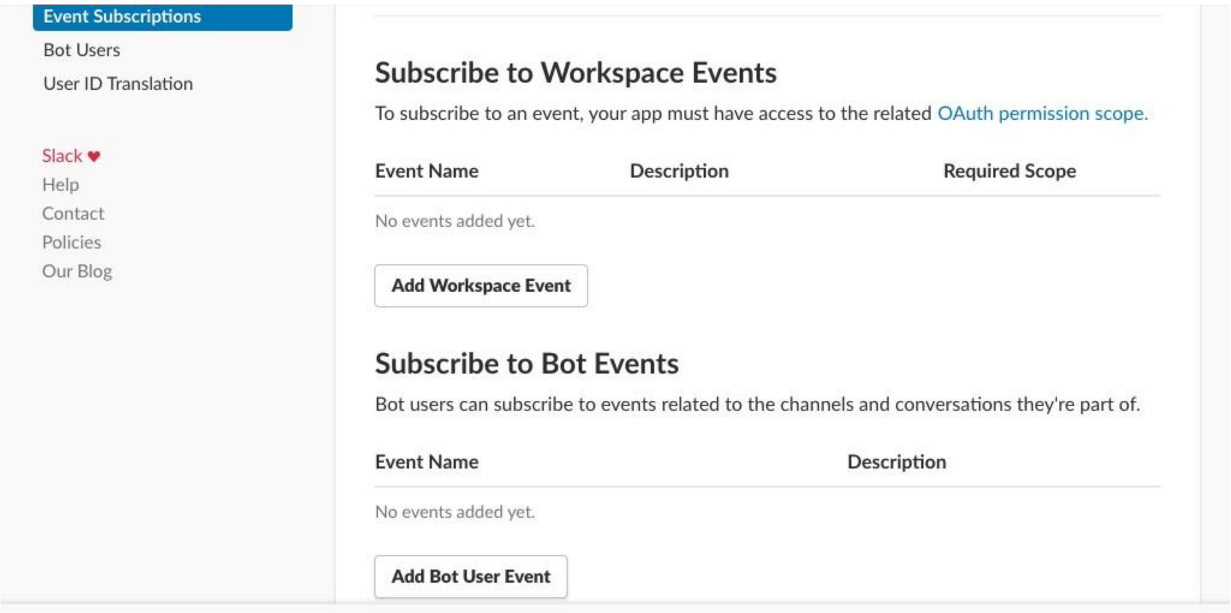
This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.



Bot User

You can bundle a bot user with your app to interact with users in a more conversational manner. Learn more about how bot users work.

3. Subscribe to events (message.im)



Subscribe to Workspace Events

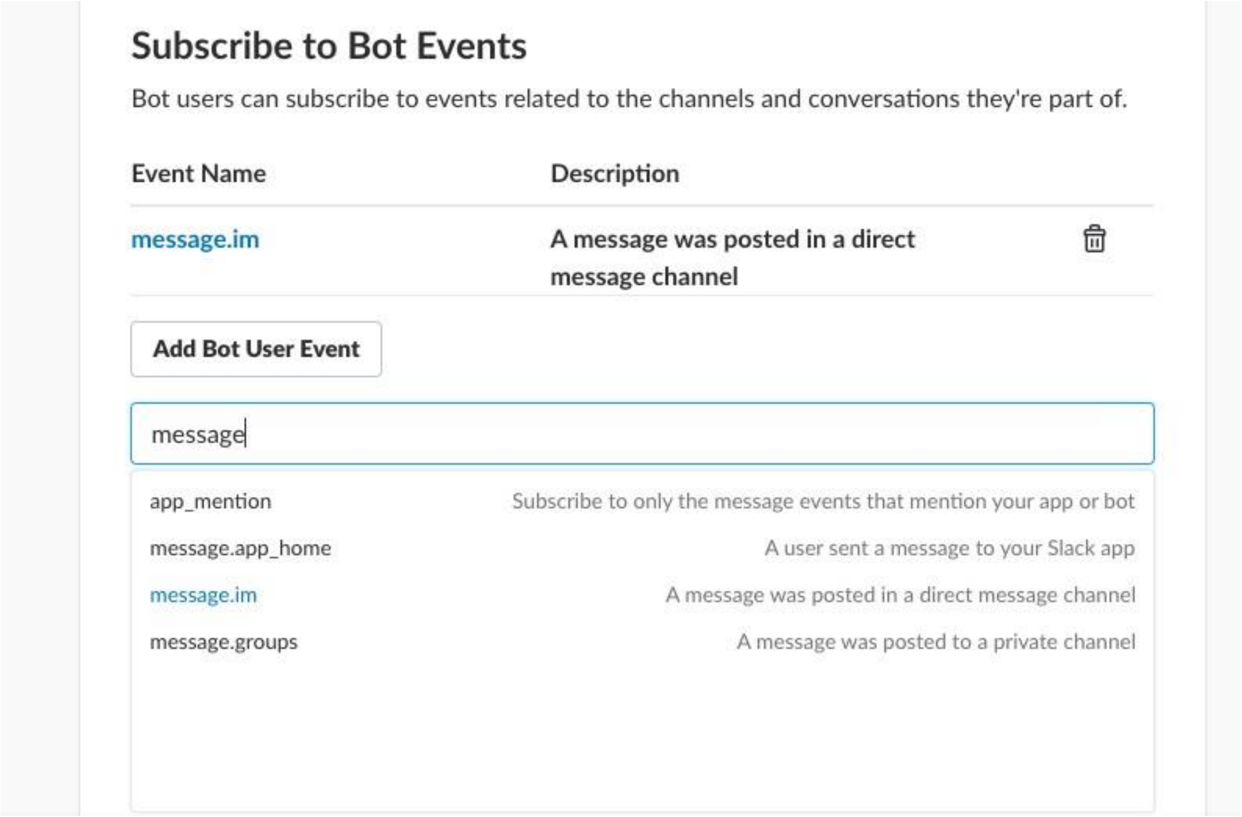
To subscribe to an event, your app must have access to the related OAuth permission scope.

Event Name	Description	Required Scope
No events added yet.		

Subscribe to Bot Events

Bot users can subscribe to events related to the channels and conversations they're part of.

Event Name	Description
No events added yet.	



Subscribe to Bot Events

Bot users can subscribe to events related to the channels and conversations they're part of.

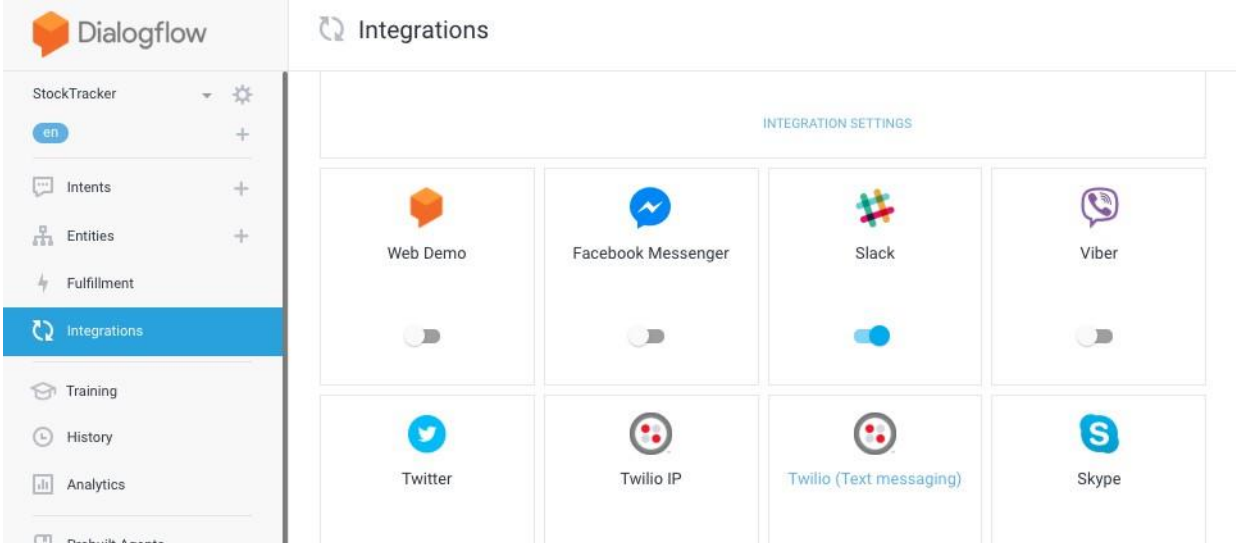
Event Name	Description
message.im	A message was posted in a direct message channel

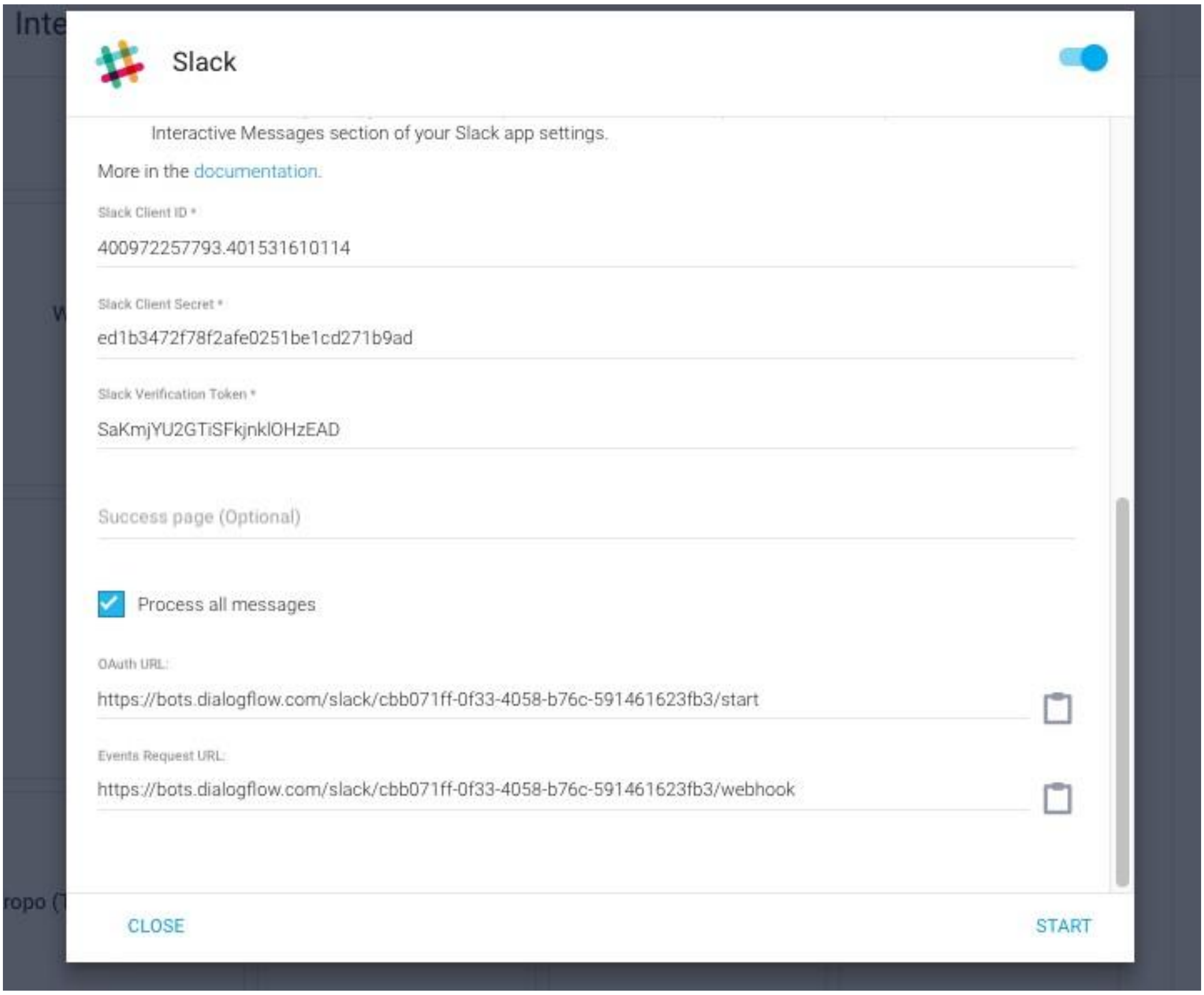
Add Bot User Event

message

app_mention	Subscribe to only the message events that mention your app or bot
message.app_home	A user sent a message to your Slack app
message.im	A message was posted in a direct message channel
message.groups	A message was posted to a private channel

4. Head over to Dialogflow:





Interactive Messages section of your Slack app settings.

More in the documentation.

Slack Client ID *

400972257793.401531610114

Slack Client Secret *

ed1b3472f78f2afe0251be1cd271b9ad

Slack Verification Token *

SaKmjYU2GTiSFkjnklOHzEAD

Success page (Optional)

☑ Process all messages

OAuth URL:

https://bots.dialogflow.com/slack/cbb071ff-0f33-4058-b76c-591461623fb3/start

Events Request URL:

https://bots.dialogflow.com/slack/cbb071ff-0f33-4058-b76c-591461623fb3/webhook

copy the oauth & event URLs

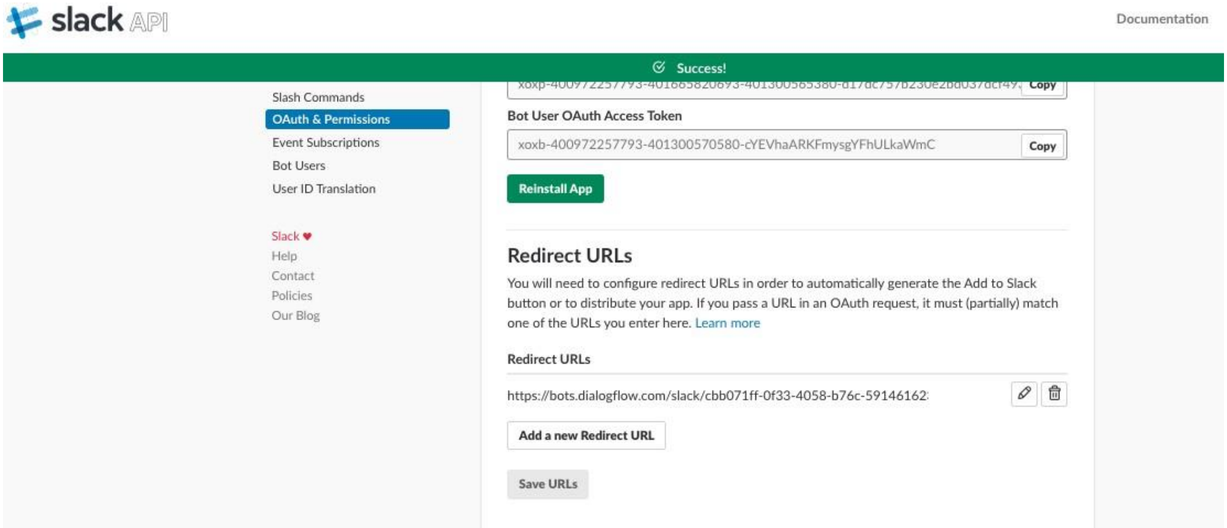
oAuth

https://bots.dialogflow.com/slack/cbb071ff-0f33-4058-b76c-591461623fb3/start

event

https://bots.dialogflow.com/slack/cbb071ff-0f33-4058-b76c-591461623fb3/webhook

5. Go back to Slack API…add oAuth url & Event Subscription URLs



xoxp-400972257793-401665820693-401300565380-d17dc757b230e2bd037dcf49

Bot User OAuth Access Token

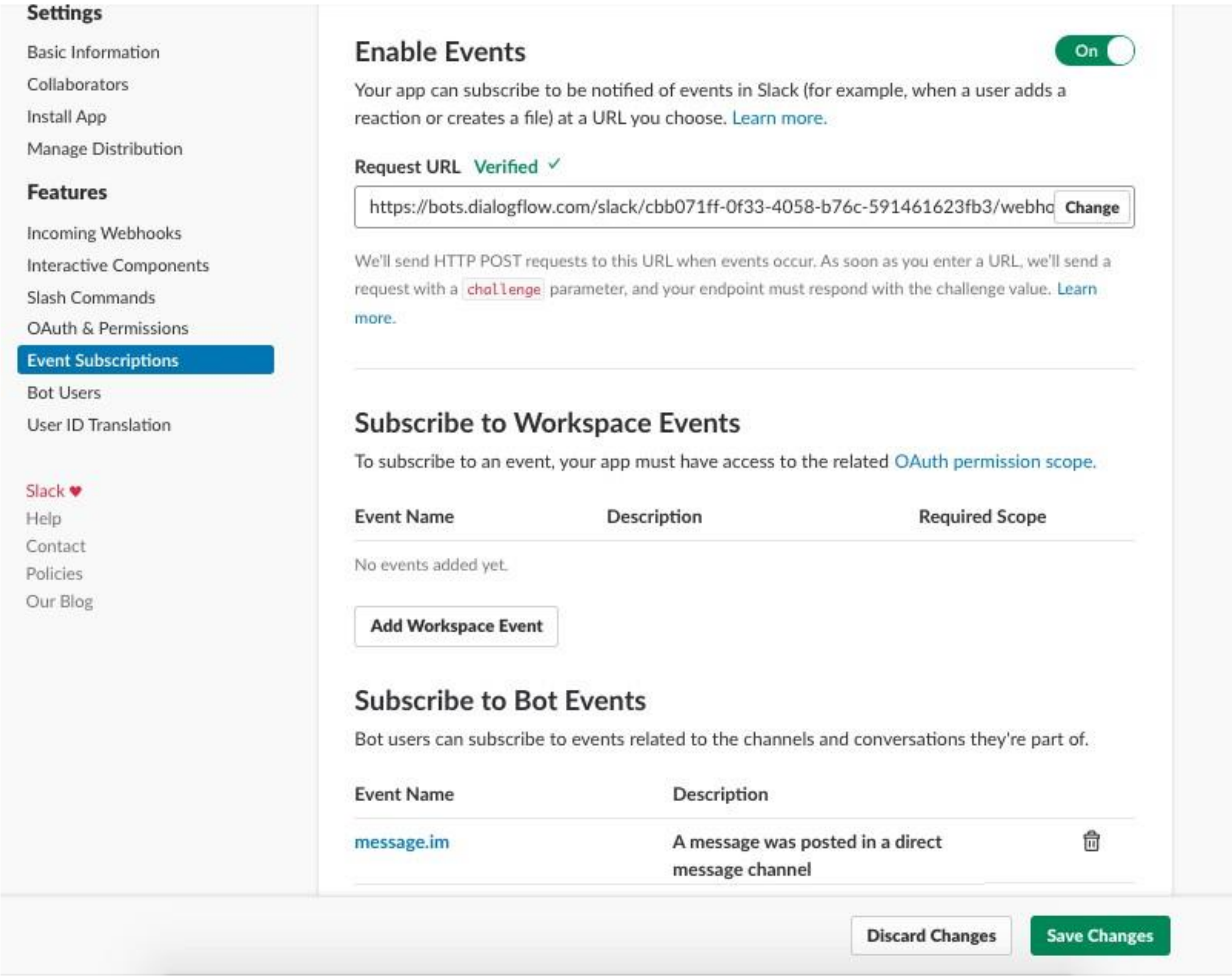
xoxb-400972257793-401300570580-cYEVhaARKFmysgYFhULkaWmC

Redirect URLs

You will need to configure redirect URLs in order to automatically generate the Add to Slack button or to distribute your app. If you pass a URL in an OAuth request, it must (partially) match one of the URLs you enter here. Learn more

Redirect URLs

https://bots.dialogflow.com/slack/cbb071ff-0f33-4058-b76c-59146162



Enable Events

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. Learn more.

Request URL Verified ✓

https://bots.dialogflow.com/slack/cbb071ff-0f33-4058-b76c-591461623fb3/webho

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. Learn more.

Subscribe to Workspace Events

To subscribe to an event, your app must have access to the related OAuth permission scope.

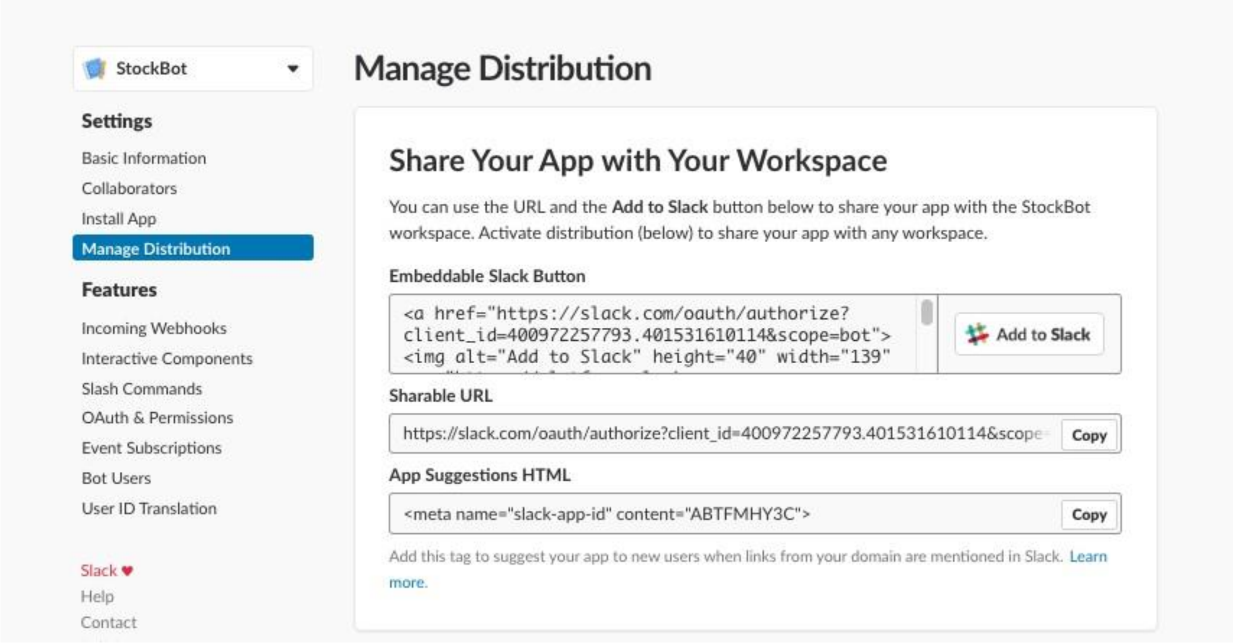
Event Name	Description	Required Scope
No events added yet.		

Subscribe to Bot Events

Bot users can subscribe to events related to the channels and conversations they're part of.

Event Name	Description
message.im	A message was posted in a direct message channel

6. Distribution:



Manage Distribution

Share Your App with Your Workspace

You can use the URL and the **Add to Slack** button below to share your app with the StockBot workspace. Activate distribution (below) to share your app with any workspace.

Embeddable Slack Button

```
<a href="https://slack.com/oauth/authorize?
client_id=400972257793.401531610114&scope=bot">
<img alt="Add to Slack" height="40" width="139"
```

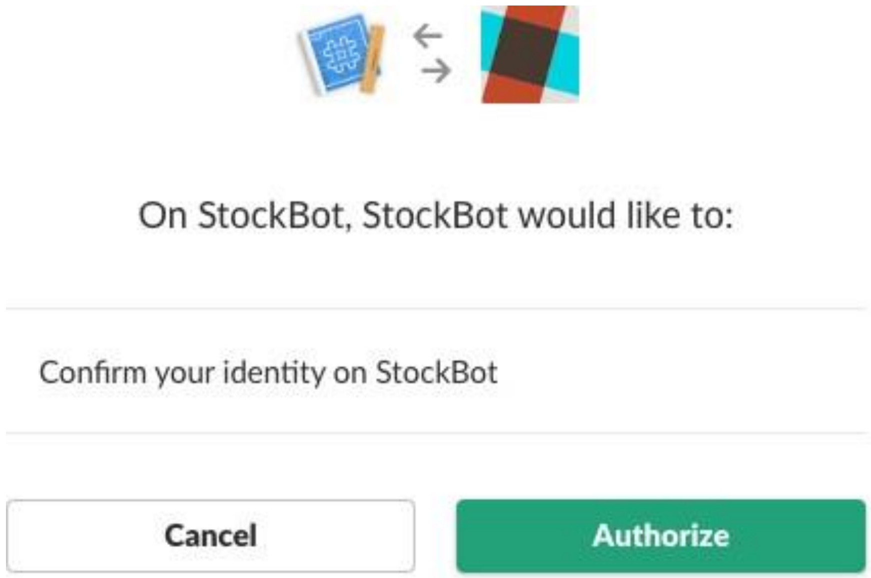
Sharable URL

https://slack.com/oauth/authorize?client_id=400972257793.401531610114&scope=

App Suggestions HTML

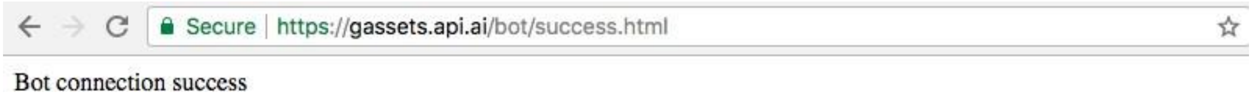
<meta name="slack-app-id" content="ABTFMHY3C">

Add this tag to suggest your app to new users when links from your domain are mentioned in Slack. Learn more.



On StockBot, StockBot would like to:

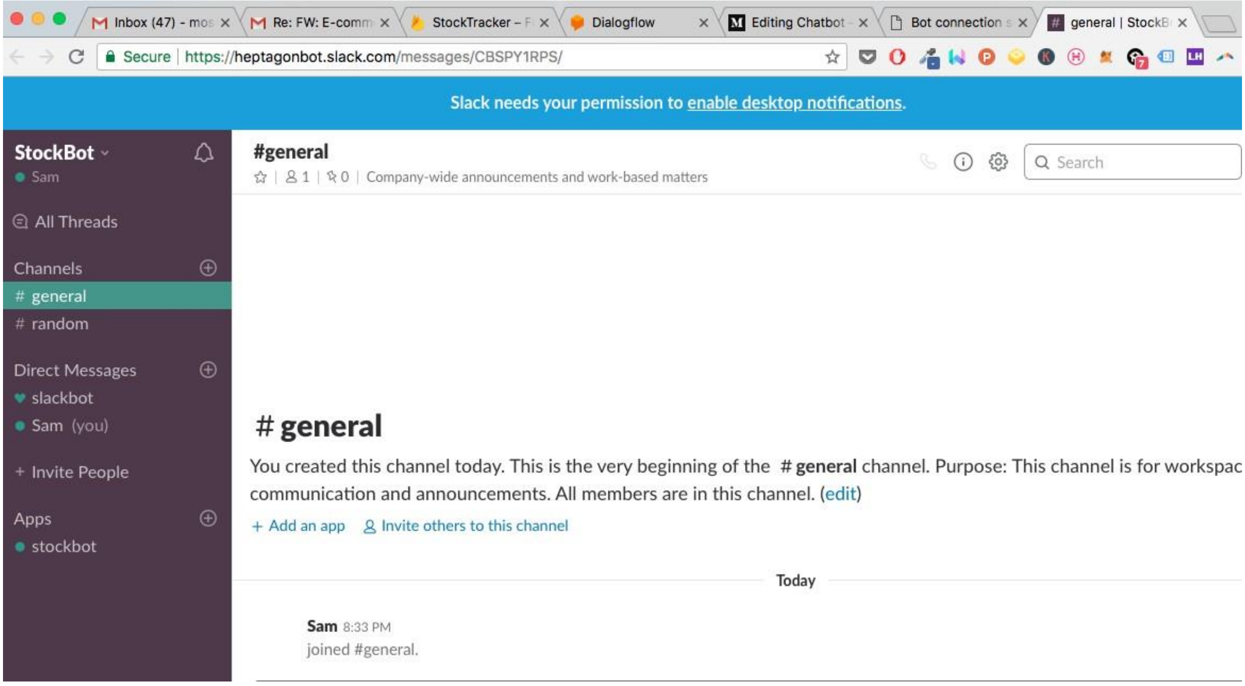
Confirm your identity on StockBot



Secure | https://gassets.api.ai/bot/success.html

Bot connection success

7. Test: Now let’s go back to our slack workspace and take the bot for a test spin.

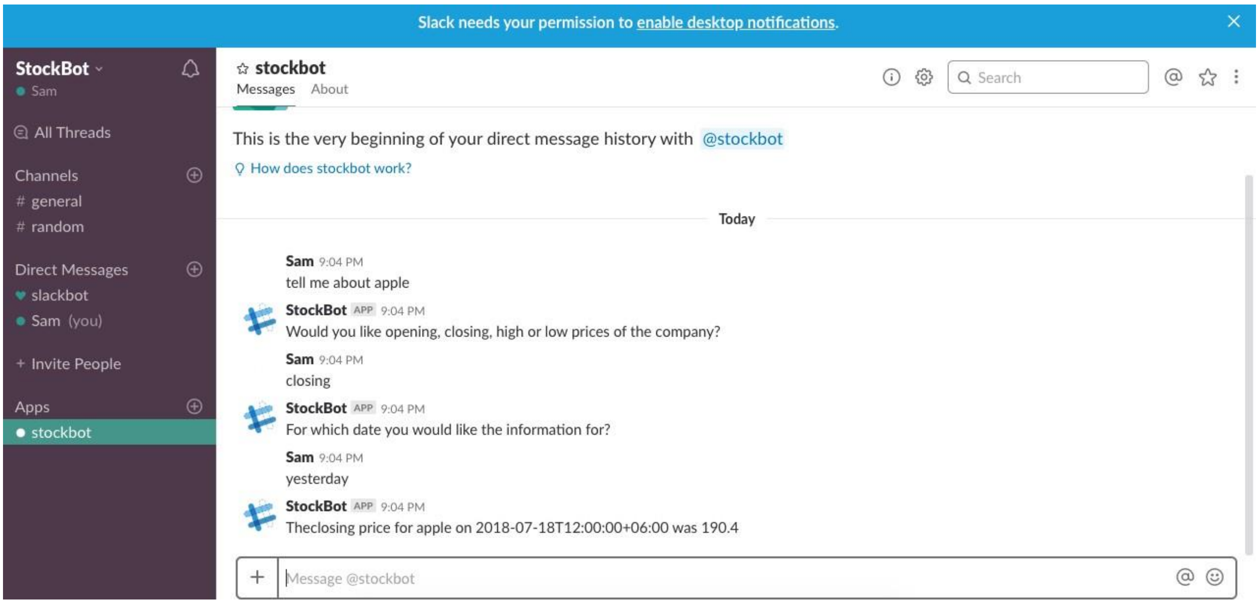


Slack needs your permission to enable desktop notifications.

general

You created this channel today. This is the very beginning of the # general channel. Purpose: This channel is for workspac communication and announcements. All members are in this channel. (edit)

Sam 8:33 PM
joined #general.



This is the very beginning of your direct message history with @stockbot

How does stockbot work?

Today

Sam 9:04 PM
tell me about apple

StockBot APP 9:04 PM
Would you like opening, closing, high or low prices of the company?

Sam 9:04 PM
closing

StockBot APP 9:04 PM
For which date you would like the information for?

Sam 9:04 PM
yesterday

StockBot APP 9:04 PM
Theclosing price for apple on 2018-07-18T12:00:00+06:00 was 190.4