



Maintaining Context and Taking Actions



Agenda

**Maintaining Context in Your
Conversation**

Taking Actions With Fulfilment



The importance of context

What about tomorrow?

Contexts allow the agent to keep the continuity in the dialog.

Context allows the agent to avoid repetition in a conversation.



Contexts allow the agent to keep track of where the user is in the conversation

Can I have a large pepperoni pizza to be picked up in one hour?

Yes, your order has been placed.

Actually, make that 2 hours.



Input and output contexts tell Dialogflow what to do

Input context

Match the intent only if the user utterance is a close match and if the context is active.

Output context

Activate a context (if it's not already active) or to maintain the context active after the intent is matched.



Demo

Adding Context to an Intent

In this demo, we'll create two new intents for capturing negative and positive feedback, and add context to them.



Best practices: When and how to use context

yes

SAVE



Define synonyms ?



Allow automated expansion

Yes

Yes, yes, please, please, yeah

no

SAVE



Define synonyms ?



Allow automated expansion

no

no, nah, no, thank you, nope, forget it, never

[Click here to edit entry](#)



Use context or follow-up intents

Context

● order.pizza
● order.pizza.upsell-drink - no
● order.pizza.upsell-drink - yes

Follow-up Intent

● query -- 1 ^
● ↳ query -- 1 - no
● ↳ query -- 1 - yes



Agenda

Maintaining Context in Your Conversation

Taking Actions With Fulfilment



Take action with Fulfillment

1

Connect to your business logic.

2


Generate dynamic responses.

3

Make stuff happen outside Dialogflow.



Creating Fulfillment

 Dialogflow

order-chatbot

en

Intents

Entities



Knowledge ^[beta]

Fulfillment


Integrations

Training

History

  Fulfillment


Webhook

DISABLED 


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)

ENABLED 

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. [Docs](#)

index.js package.json 

```
1 // See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
2 // for Dialogflow fulfillment library docs, samples, and to report issues
3 'use strict';
4
5 const functions = require('firebase-functions');
6 const {WebhookClient} = require('dialogflow-fulfillment');
7 const {Card, Suggestion} = require('dialogflow-fulfillment');
8
9 process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements
10
```



Don't use follow-up intents to capture missing entities

●	↳	some intent - no	^
●	↳	some intent - no - yes	^
●	↳	some intent - no - yes - more	
●	↳	some intent - no - later	^
●	↳	some intent - no - later - cancel	
●	↳	some intent - custom	^
●	↳	some intent - custom - cancel	^
●	↳	some intent - custom - cancel - yes	



Use slot filling instead

order.pizza

SAVE

Action and parameters

Enter action name

REQUIRED ?	PARAMETER NAME ?	ENTITY	PROMPTS
<input checked="" type="checkbox"/>	pizza_topping	@pizza_topping	1 Yum! What toppings would you like?
<input type="checkbox"/>	number	@s	2 What toppings would you like on your pizza?
<input checked="" type="checkbox"/>	size	@s	3 Enter a prompt variant
<input checked="" type="checkbox"/>	time	@s	
<input type="checkbox"/>	Enter name	Enter entity	Enter value

+ New parameter

Prompts for "pizza_topping"

NAME	ENTITY	VALUE
pizza_topping	@pizza_topping	\$pizza_topping

PROMPTS

- 1 Yum! What toppings would you like?
- 2 What toppings would you like on your pizza?
- 3 Enter a prompt variant

CLOSE



Demo

Adding Fulfillment

In this demo, we'll write a Cloud Function to do fulfillment.



Lab

Lab 1,Part 2



cloud.google.com

