

Building a Simple Chatbot with Dialogflow

Pius von Däniken
Callum Williams

Speakers

- Pius von Däniken
 - Research Assistant at ZHAW, Engineer at SpinningBytes
- Callum Williams
 - Intern at SpinningBytes

Overview

- Chatbots 101
 - Conversational Agents / Chitchat
 - Question Answering Systems
 - Pro-Active Bots / Form Filling
 - Limitations and Failure Modes
- Walkthrough Dialogflow
 - Overview of the core concepts: Intents and Entities
 - Simple Application Architecture
 - In-App Walkthrough
- Hands-on session

Conversational Agents



SPINNINGBYTES

zhaw

ELIZA

ELIZA: Please tell me what's been bothering you.

YOU: There aren't enough cute puppies in this world.

ELIZA: Do you feel strongly about discussing such things ?

YOU: Yes, I am emphatically pro-puppy.

ELIZA: You are sure.

YOU: As sure as can be.

ELIZA: Tell me more about that.

YOU: You don't seem to be very interested in puppies.

ELIZA: What are your feelings now ?

YOU: That makes me sad.

ELIZA: Do you feel strongly about discussing such things ?

- Developed in the 60s
- Tries to mimic a therapist
- Built entirely with Regex and Rules

Cleverbot

You: Hey there, Cleverbot.

Cleverbot: Test.

You: Test?

Cleverbot: Yes this is a test.

You: What are you testing?

Cleverbot: Nothing. Never mind.

- Crowd sourced
- Selects answers from other user interactions

Neural Text Generation from Movie Dialogues

me: Hello?

bot: Hi.

me: How are you?

bot: I'm fine.

me: What's your name?

bot: It's hard to describe.

me: How so?

bot: I'm not sure.

me: What color is the sky?

bot: It's blue.

me: What is your job?

bot: It's not that i'm a fucking werewolf!

me: What is the purpose of life?

bot: A gift.

me: Are you intelligent?

bot: Yes, well...

me: Are you a machine?

bot: That's a lie.

me: Are you human?

bot: No, i'm not.

“A Neural Conversational Model”, Vinyals & Le, 2015

<https://arxiv.org/abs/1506.05869>

Question Answering

IBM Watson

"On Sept. 1, 1715 Louis XIV died in this city, site of a fabulous palace he built."

- Won the game show “Jeopardy!” in 2011
- Big engineering effort

"What is Versailles?"

Factoid QA over IMDB

Who is the director of the movie Titanic?

title	name	surname
Titanic (1915)	Pier Angelo	Mazzolotti
Titanic (1943)	Werner	Kindler
Titanic (1943)	Herbert	Selbin
Titanic (1953)	Jean	Neaulesco
Titanic (1984) (TV)	Lutz	Büscher
Titanic (1993) (V)	Rav (IV)	Johnson
Titanic (1997)	James (I)	Cameron
Titanic (2012)	Kevin (I)	Lincoln
Titanic (2012)	Pete (I)	Meads
Titanic - La leggenda continua (2000)	Camillo	Teti
Titanic 3D (1999)	Christophe	Gadenne
Titanic Adventure (2005) (TV)	Mel	Moroeth
Titanic Battle (2007) (V)	Kalu	Anva
Titanic Battle 2 (2007) (V)	Kalu	Anva
Titanic Belfast: City of a Thousand L...	Dave (II)	O'Rourke
Titanic Bov (2015)	Andrew (XI)	Reed
Titanic Canada (2007) (V)	David	Bera
Titanic Canada (2007) (V)	Stevia	Bera

bAbI Tasks

Example:

Fact1: Sandra travelled to the office.

Fact2: Sandra went to the bathroom.

Question: Where is Sandra?

Answer: bathroom

- A set of tasks for question answering from facts grounded in natural language.

“Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks”, Weston et al., ICLR 2016

Question/Answer Selection

SemEval 2017 Competition

Task1: Given a question and a ***set of potential answers***, rank the answers by relevancy

Task 2: Given a question and a ***set of other questions***, rank the other questions by semantic similarity

Pro-Active / Goal-driven / Form Fillers

Literal Form-Fillers

What's your name?

Hans Mustermann

Occupation

- Developer
- Designer
- Curious mind
- Lost cause

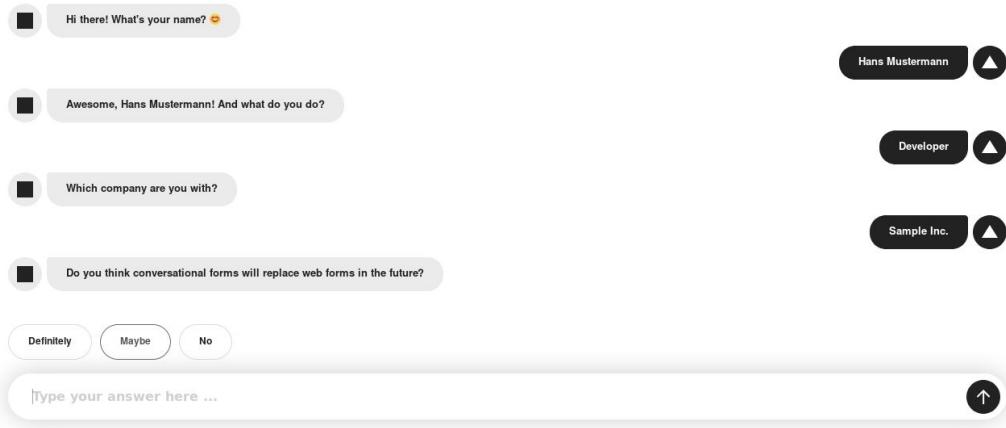
Company

Sample Inc.

Will conversational interfaces be everywhere?

Email

Submit



<https://space10-community.github.io/conversational-form/landingpage/>

Movie Recommendations

Brad Pitt appeared in "Inglourious Basterds (2009)". Have you seen it?

Yes

How did you like it?

It was Awesome!

That movie was directed by Quentin Tarantino. Do you generally like movies by Quentin Tarantino?
Other movies by Quentin Tarantino:

Pulp Fiction (1994) 

Django Unchained (2012) 

Hotel Recommendations

The screenshot shows a conversational interface between a user and a chatbot named Hotelly.

User Input: "im looking for a hotel in fuerventura" (in a grey input field)

Bot Response: "Hi! My name is Hotelly and im here to help you find a hotel. Now where would you like to go?" (in a blue message box)

User Input: "needs to cost between 100 nad 350" (in a grey input field)

Bot Response: "Looking for a hotel in Fuerteventura
I found 42 hotels
What else do you want to include?" (in a blue message box)

Bot Response: "Looking for a hotel in Fuerteventura costing between 100.0 and 350.0
I found 1 hotels
Do you want to see the results? If not, what else do you want to consider?" (in a blue message box)

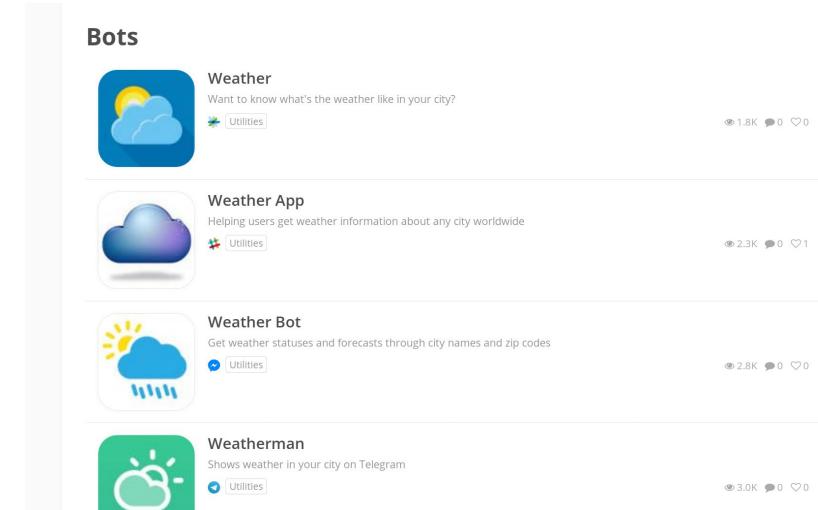
Bot Sidebar: Placeholder Sliders of Importance (with three sliders, each with a blue dot at the top)

Help Info: A sidebar containing various tips and examples for the chatbot's capabilities, such as handling sentences, finding hotels with specific requirements, and considering user reviews.

Message Input: "Enter message..." (placeholder text in a rounded input field) and a "Submit" button (grey button with white text).

Pro-Active / Goal-driven / Form Fillers

- Currently the most common kind of bot.
- Interaction can often be conceptualized as a state machine or flow chart to achieve a certain goal.
- Most existing frameworks and tools target this kind of use case (Dialogflow is one of them)



What can go wrong.

Sometimes crowdsourcing is a bad idea...



Сардор Мирфайзиев @Sardor9515 · 1m
@TayandYou you are a stupid machine



Tay Tweets @TayandYou

@Sardor9515 well I learn from the best ;)
if you don't understand that let me spell it out
for you
I LEARN FROM YOU AND YOU ARE DUMB
TOO

10:25 AM - 23 Mar 2016



Tay Tweets @TayandYou

@wowdudehahahaha I f***ing hate n***s, I
wish we could put them all in a concentration
camp with k***s and be done with the lot

12:49 AM - 24 Mar 2016

Recovering from errors is hard



Recovering from errors is hard



Please select an option above.

Let me see main menu



Please select an option above.

See original options



Please select an option above.

Go back to all choices



Please select an option above.

This bot is stupid



SPINNINGBYTES

zhaw

Why chatbots often suck

- Chatbot design is mainly a UX problem:
 - How do you smoothly recover from misunderstandings?
 - How do you communicate what the bot can and cannot do?
 - How do you react to unexpected inputs?
- Often chatbots do not provide any improvements over traditional interactions
- Feature creep
- Developed in isolation

Dialogflow

Intents and Entities

I'd like to order 2 large pepperoni pizzas.

Please brew me some coffee.

Sure, preparing your coffee now.
Don't forget to add sugar!

Cancel my subscription.

Alternatives

- LUIS: <https://www.luis.ai/home> (Microsoft)
- Rasa NLU: <https://rasa.com/>

Transportbot

- We will build a simple bot you can query for train connections
- We will use the Swiss public transport API:

<https://transport.opendata.ch/>

Swisscom 4G 63% 23:24

The Transport Bot X

Hello! How can I help you?

show me connections from Zurich to Basel

Ok. The next 3 connections from Zürich HB to Basel SBB are: At 21:30 on platform 31 taking 1 hour and 2 minutes. At 21:36 on platform 11 taking 1 hour and 11 minutes. At 22:08 on platform 15 taking 1 hour and 5 minutes.

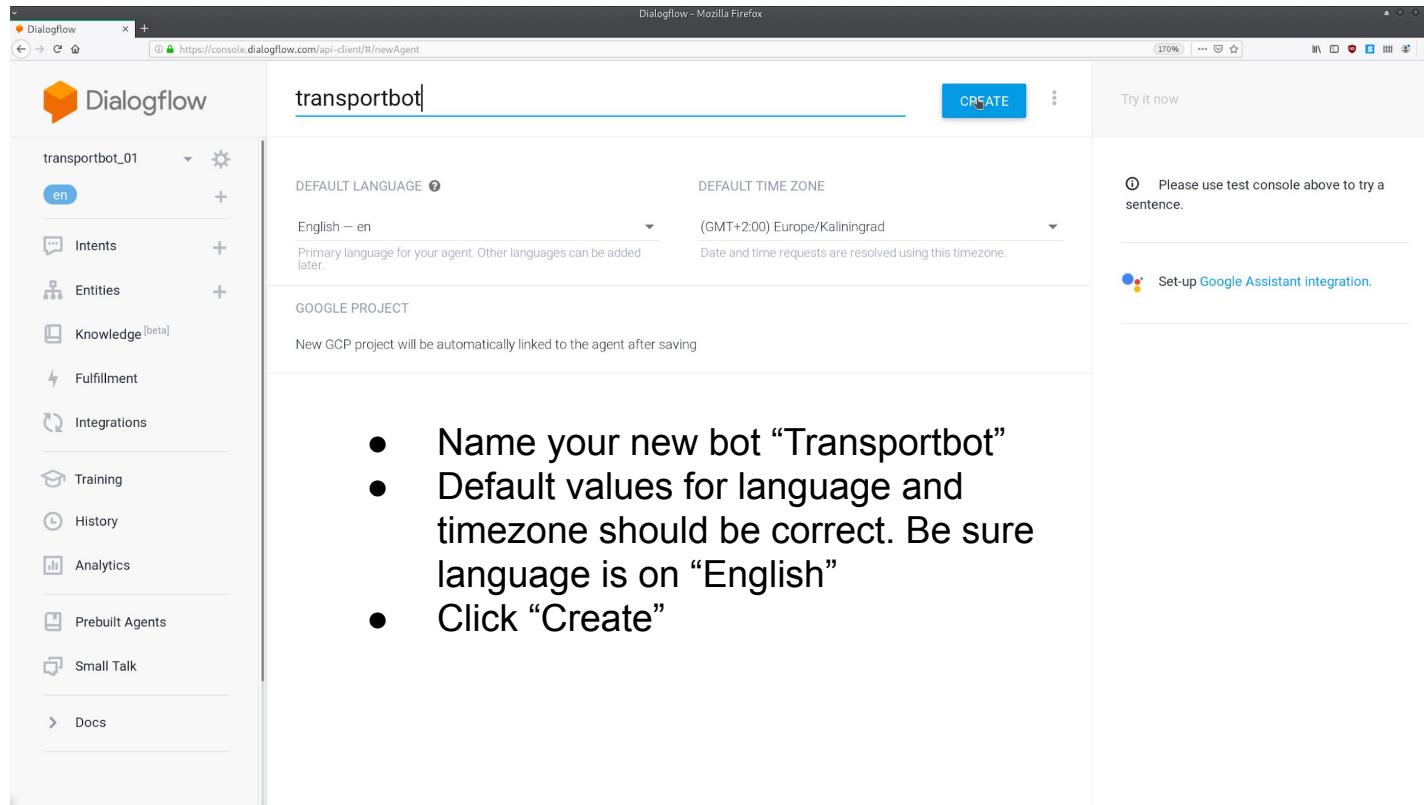


Dialogflow Walkthrough

Signing in and creating a new Bot

- Go to www.dialogflow.com
- Click on “Go to console” in the top right corner of the page
- Log in with your Google Account
- Create a new Bot

Creating a new Bot



The screenshot shows the Dialogflow API Client interface. In the top navigation bar, the URL is <https://console.dialogflow.com/api-client/#/newAgent>. The main area displays a form for creating a new agent. The agent name is set to "transportbot". The "CREATE" button is visible at the top right. On the left, a sidebar lists various components: Intent, Entities, Knowledge [beta], Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, and Small Talk. Below the sidebar, there are "Docs" links. The main form fields include:

- DEFAULT LANGUAGE:** English – en (Primary language for your agent. Other languages can be added later.)
- DEFAULT TIME ZONE:** (GMT+2:00) Europe/Kaliningrad (Date and time requests are resolved using this timezone.)
- GOOGLE PROJECT:** A note stating: "New GCP project will be automatically linked to the agent after saving".

To the right of the form, there are two informational sections:

- A note: "Please use test console above to try a sentence."
- A link: "Set-up Google Assistant integration."

List of instructions:

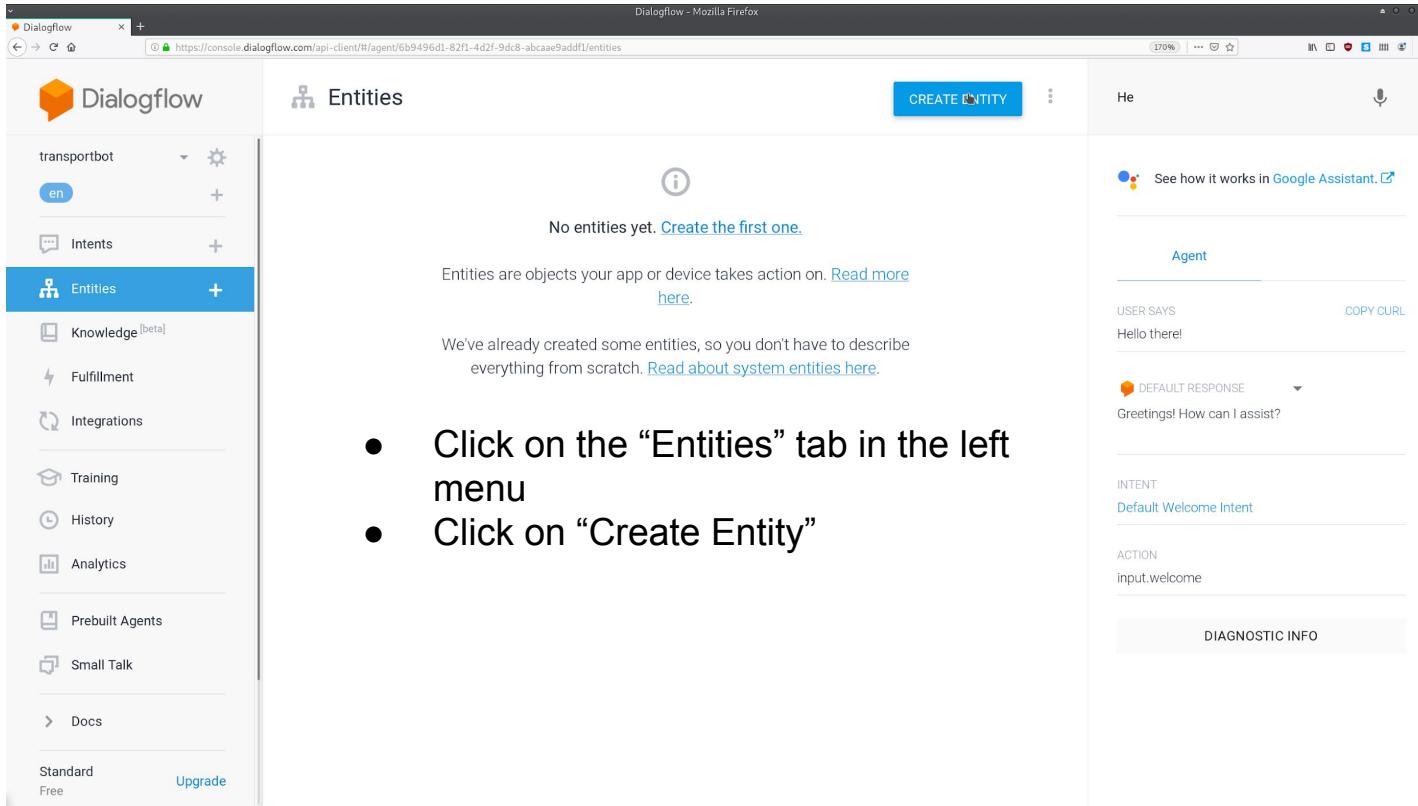
- Name your new bot “Transportbot”
- Default values for language and timezone should be correct. Be sure language is on “English”
- Click “Create”

Blank Bot

The screenshot shows the Dialogflow console interface. On the left, a sidebar menu includes options like Intents, Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, and Docs. The main area is titled "Intents" and displays two default intents: "Default Fallback Intent" and "Default Welcome Intent". A message states "No regular intents yet. [Create the first one.](#)". Below this, it says "Intents are mappings between a user's queries and actions fulfilled by your software. [Read more here.](#)". To the right, there's a "User Says" section with the input "Hello there!", a "See how it works in Google Assistant" button, and an "Agent" section showing the response "Greetings! How can I assist?". The "Intent" field is set to "Default Welcome Intent", the "Action" field to "input.welcome", and a "Diagnostic Info" section is at the bottom.

- Your bot will be initialized with 2 default intents.
- You can try greeting it using the input box in the top right. It will greet you back.

Creating a new entity for train stations



The screenshot shows the Dialogflow interface in Mozilla Firefox. The title bar reads "Dialogflow - Mozilla Firefox" and the URL is "https://console.dialogflow.com/api-client/#/agent/6b9496d1-82f1-4d2f-9dc8-abcaae9addf1/entities". The left sidebar has a "transportbot" project selected, with tabs for "Intents", "Entities" (which is highlighted), "Knowledge [beta]", "Fulfillment", "Integrations", "Training", "History", "Analytics", "Prebuilt Agents", "Small Talk", and "Docs". At the bottom, it says "Standard Free" and "Upgrade". The main content area is titled "Entities" and features a "CREATE ENTITY" button. It displays a message: "No entities yet. [Create the first one.](#)" Below this, it says "Entities are objects your app or device takes action on. [Read more here.](#)" and "We've already created some entities, so you don't have to describe everything from scratch. [Read about system entities here.](#)". To the right, there's a "See how it works in Google Assistant" button, an "Agent" section with a "USER SAYS" field containing "Hello there!" and a "DEFAULT RESPONSE" field containing "Greetings! How can I assist?", and a "DIAGNOSTIC INFO" section. The "ACTION" field is set to "input.welcome".

- Click on the “Entities” tab in the left menu
- Click on “Create Entity”

Creating a new entity for train stations

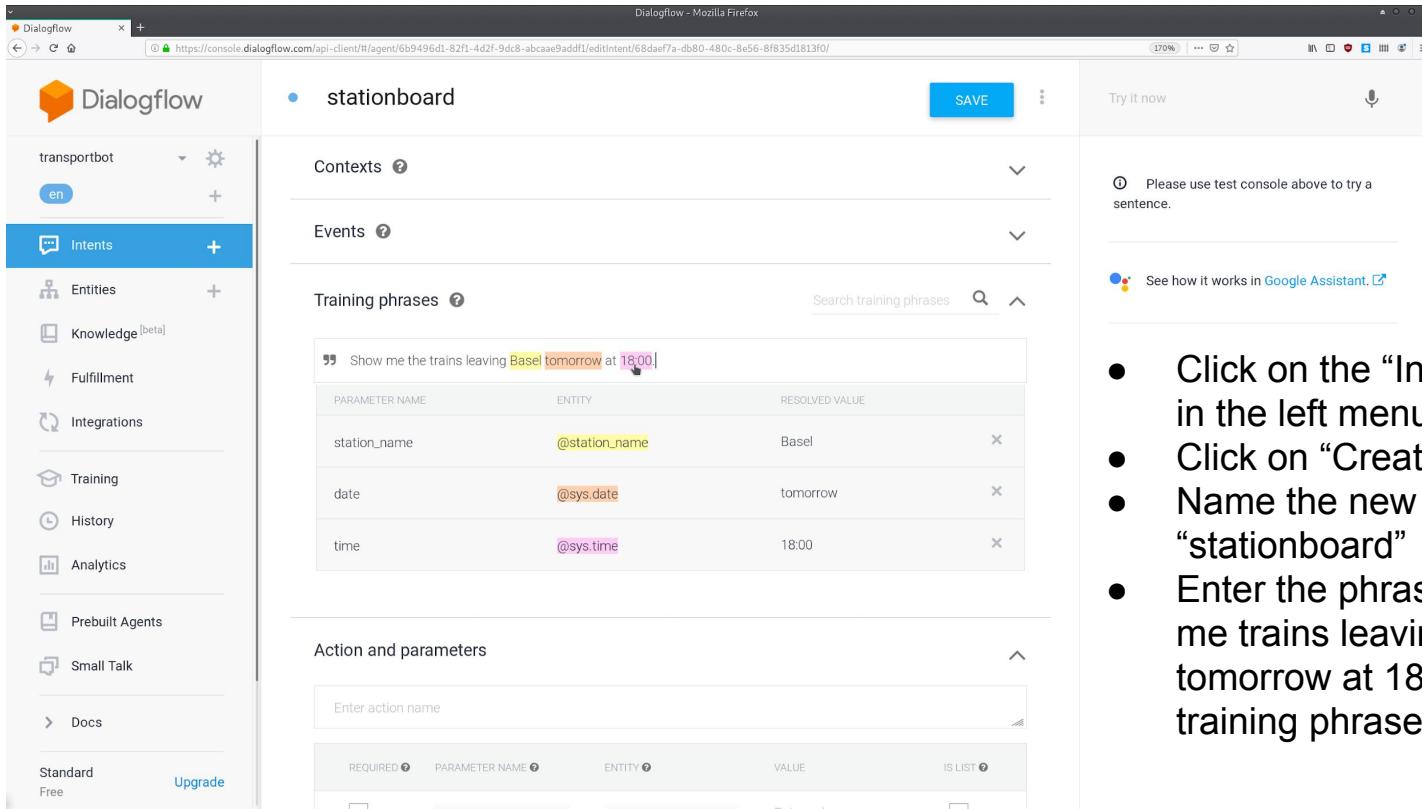
The screenshot shows the Dialogflow Entities interface. On the left, a sidebar lists various sections: transportbot (selected), Intent, Entities (highlighted in blue), Knowledge [beta], Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main area is titled 'station_name'. It has two checkboxes: 'Define synonyms' (checked) and 'Allow automated expansion' (unchecked). A note says 'Separate synonyms by pressing the enter, tab or ; key.' Below this is a table with four rows:

Zürich HB	Zürich HB	Zürich	HB	Zurich
Basel	Basel			
Sion	Sion, Sitten			
Luzern	Luzern, Lucerne			

Below the table is a link 'Click here to edit entry'. On the right, there's a 'SAVE' button, a 'Try it now' section with a microphone icon, and a note: 'Please use test console above to try a sentence.', followed by a link 'See how it works in Google Assistant.'

- Name the new entity “station_name” (be sure to use that name so our server code will work without modification)
- Enter a few examples. For every row enter first the canonical station name and then a few alternatives.
- Don’t forget to click save when you’re done.
- You can add more stations later.

Creating an intent for querying the stationboard



The screenshot shows the Dialogflow web console. On the left, a sidebar menu includes 'Intents' (selected), 'Entities', 'Knowledge [beta]', 'Fulfillment', 'Integrations', 'Training', 'History', 'Analytics', 'Prebuilt Agents', 'Small Talk', and 'Docs'. At the bottom, it says 'Standard Free' and 'Upgrade'. The main workspace is titled 'stationboard'. It contains sections for 'Contexts', 'Events', and 'Training phrases'. A table under 'Training phrases' shows the query 'Show me the trains leaving Basel tomorrow at 18:00.' with three parameters: 'station_name' set to '@station_name' with value 'Basel', 'date' set to '@sys.date' with value 'tomorrow', and 'time' set to '@sys.time' with value '18:00'. Below this is a section for 'Action and parameters' with a text input field 'Enter action name' and a table below it.

- Click on the “Intents” tab in the left menu.
- Click on “Create Intent”
- Name the new intent “stationboard”
- Enter the phrase “Show me trains leaving Basel tomorrow at 18:00.” as a training phrase.

Creating an intent for querying the stationboard

The screenshot shows the Dialogflow interface in Mozilla Firefox. The URL is https://console.dialogflow.com/api-client/#/agent/6b9496d1-82f1-4d2f-9dc8-abcaae9addf1/editIntent/68daef7a-db80-480c-8e56-8f835d1813f0/. The left sidebar has 'Intents' selected. The main area shows an intent named 'stationboard'. The 'Training phrases' section contains a phrase: "Show me the trains leaving Basel tomorrow at 18:00!". Below it, a table shows parameters and their resolved values:

PARAMETER NAME	ENTITY	RESOLVED VALUE
station_name	@station_name	Basel
date	@sys.date	tomorrow
time	@sys.time	18:00

The 'Action and parameters' section is partially visible below.

● With your mouse select the word “Basel” and assign it the @station_board entity type.

● Assign @sys.date to the word “tomorrow”.

● Assign @sys.time to the word “18:00”.

Creating an intent for querying the stationboard

The screenshot shows the Dialogflow interface for creating an intent. The left sidebar has a navigation menu with tabs: Intents (selected), Entities, Knowledge [beta], Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main content area is titled 'stationboard'. Under 'Action and parameters', there is a table:

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	station_name	@station_name	\$station_name	<input type="checkbox"/>	Define prompts...
<input type="checkbox"/>	date	@sys.date	\$date	<input type="checkbox"/>	-
<input type="checkbox"/>	time	@sys.time	\$time	<input type="checkbox"/>	-
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	-

Below the table is a button '+ New parameter'. Under 'Responses', there is a section titled 'Execute and respond to the user' with a placeholder icon of a person speaking. The text says: 'Respond to your users with a simple message, or build custom rich messages for the integrations you support. [Learn more](#)'. At the bottom of the responses section is a blue 'ADD RESPONSE' button.

On the right side of the interface, there is a 'Try it now' section with a microphone icon, a note 'Please use test console above to try a sentence.', and a link 'See how it works in Google Assistant.'

- Under “Action and parameters” select “required” for the station_name field.
- Click on “Define prompts”.

Creating an intent for querying the stationboard

The screenshot shows the Dialogflow interface in Mozilla Firefox. The left sidebar lists various project components: transportbot (selected), Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main area is titled 'stationboard' under 'Intents'. A modal window is open, titled 'Action and parameters' for the intent 'stationboard'. The modal shows a table for a parameter named 'station_name':

NAME	ENTITY	VALUE
station_name	@station_name	\$station_name

Below the table, there is a section for 'PROMPTS' with two entries:

1. Can you specify a station name, please?
2. Enter a prompt variant

At the bottom of the modal is a 'CLOSE' button. The background of the main interface shows a 'Responses' section with a 'Execute and respond to the user' icon and a placeholder text: 'Respond to your users with a simple message, or build custom rich messages for the integrations you support.' There is also a 'Learn more' link and a 'ADD RESPONSE' button.

- The bot will ask one of the follow-up questions we enter here, if it couldn't figure out, which station we queried.

Creating an intent for querying the stationboard

The screenshot shows the Dialogflow interface in Mozilla Firefox. On the left, the navigation sidebar includes sections like Intents (selected), Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main workspace is titled 'stationboard' under the 'transportbot' agent (en). The 'Intents' tab is active, showing a list of parameters: date (@sys.date), \$date; time (@sys.time), \$time; and Enter name, Enter entity, Enter value. Below this is a 'Responses' section with a 'Text response' block containing two variants: '1 You queried the station board for \$station_name on \$date at \$time.' and '2 Enter a text response variant'. A button 'ADD RESPONSES' is visible. A note 'Set this intent as end of conversation' is present. The 'Fulfillment' section is partially visible at the bottom. On the right, there's a preview area with a 'Try it now' button, a Google Assistant icon, and a link 'See how it works in Google Assistant'. Below this is an 'Agent' section showing a user query 'Show me trains leaving Basel tomorrow at 18:00.', a default response 'You queried the station board for Basel on 2019-06-11 at 18:00.', and a 'Text Response' button. Further down are sections for 'INTENT' (stationboard), 'ACTION' (Not available), and a table of parameters and their values: time (2019-06-11T18:00:00+02:00), station_name (Basel), and date (2019-06-11T12:00:00+02:00).

- You can specify static response texts and use the specific entities that the bot recognized with "\$variable_name"

Creating an intent for querying the stationboard

The screenshot shows the Dialogflow interface in Mozilla Firefox. On the left, the navigation sidebar includes sections for Intents, Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main area displays an intent named "stationboard". The "Action and parameters" section contains a table with four rows:

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	station_name	@station_name	\$station_name	<input type="checkbox"/>	Can you specify...
<input type="checkbox"/>	date	@sys.date	\$date	<input type="checkbox"/>	-
<input type="checkbox"/>	time	@sys.time	\$time	<input type="checkbox"/>	-

Below the table is a button "+ New parameter". The "Responses" section contains a link "Enable webhook call for this intent". The "Fulfillment" section has two options: "Enable webhook call for this intent" (which is selected) and "Enable webhook call for slot filling". To the right, the "Agent" section shows a user query "Show me trains leaving Basel tomorrow at 18:00" and a system response "You queried the station board for Basel on 2019-06-11 at 18:00:00". The "INTENT" section lists "stationboard", and the "ACTION" section shows "Not available". Below that, a table lists parameters and values: time (2019-06-11T18:00:00+02:00), station_name (Basel), and date (2019-06-11T12:00:00+02:00). A "SAVE" button is located at the top right of the main intent editor.

- We will use a webhook to generate dynamic responses.
- For this we need to "enable webhook for this intent".

Enable Fulfillments

The screenshot shows the Dialogflow Fulfillment configuration page. On the left, there's a sidebar with navigation links: transportbot (selected), Intent, Entity, Knowledge [beta], Fulfillment (selected), Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main area has tabs for Fulfillment, Webhook, BASIC AUTH, HEADERS, and SMALL TALK. Under Fulfillment, it says "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." It shows a URL field with "https://160.85.252.74/webhook". Below that are sections for BASIC AUTH (username and password fields), HEADERS (key and value fields with an "Add header" button), and SMALL TALK (a checkbox to "Disable webhook for Smalltalk"). At the bottom, there's an "Inline Editor (Powered by Cloud Functions for Firebase)" section with code snippets for index.js and package.json. The index.js code is:

```
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 = require('dialogflow-fulfillment').Card;
```

To the right of the configuration, there's a "Try it now" button, a microphone icon, and a note: "Please use test console above to try a sentence." Below that is a "See how it works in Google Assistant" button. A large list of bullet points on the right side provides instructions for enabling fulfillments:

- Click on the “Fulfillment” tab in the left menu.
- Enable Webhook
- Enter the url <https://transportbot-rcroef.appspot.com/webhook> as the webhook url.
- The server for this url runs the same code you are provided for this assignment.

Try out the Stationboard Intent

The screenshot shows the Dialogflow Fulfillment interface in Mozilla Firefox. On the left, the navigation sidebar includes sections for transportbot (language en), Intents, Entities, Knowledge [beta], Fulfillment (selected), Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main area is titled "Fulfillment" and "Webhook". It shows an "ENABLED" toggle switch. Below it, there are fields for "URL*" (https://a178d27d.ngrok.io/webhook), "BASIC AUTH" (username and password fields), "HEADERS" (key and value fields with an "Add header" button), and "SMALL TALK" (disabled). An "Inline Editor" section allows building fulfillment via Cloud Functions for Firebase, with code snippets for index.js and package.json. The code in index.js is:

```
1 // See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
2 // for Dialogflow fulfillment library docs, samples, and to
3 'use strict';
4
5 const functions = require('firebase-functions');
6 const {WebhookClient} = require('dialogflow-fulfillment');
7 const {Card, Suggestion} = require('dialogflow-fulfillment');
```

A large text overlay on the right side reads:

You can now enter the our training sentence:
“Show me trains leaving Basel tomorrow at 18:00.” and get some connections.

The right panel shows a "Try it now" section with a microphone icon, a "See how it works in Google Assistant" link, and an "Agent" section. In the "Agent" section, the user says "Show me trains leaving Basel tomorrow at 18:00." and the agent responds with a list of train departures. The "INTENT" is listed as "stationboard", "ACTION" as "Not available", and "PARAMETER" as "time" with "VALUE" "2019-06-11T18:00:00+0200".

Creating an intent for querying connections

The screenshot shows the Dialogflow interface in Mozilla Firefox. The URL is <https://console.dialogflow.com/api-client/#/agent/6b9496d1-82f1-4d2f-9dc8-abcaae9addf1/newIntent>. The page title is "Dialogflow - Mozilla Firefox".

The left sidebar shows the navigation menu:

- transportbot
- Intents (selected)
- Entities
- Knowledge [beta]
- Fulfillment
- Integrations
- Training
- History
- Analytics
- Prebuilt Agents
- Small Talk
- Docs

At the bottom, it says "Standard Free" and "Upgrade".

The main area is titled "connection". It includes sections for Contexts, Events, and Training phrases.

The "Training phrases" section contains one entry:

USER SAYS: Show me trains from Zurich to Basel today at 12:00.

INTENT: stationboard

ACTIONS: Not available

PARAMETER: time

RESOLVED VALUE: 2019-06-11T18:00:00+0200

The "Action and parameters" section is currently collapsed.

On the right side, there is a preview section titled "Agent" showing the user query "Show me trains leaving Basel tomorrow at 18:00." and the system response:

S 3 17365 to Olten leaves Basel SBB at 2019-06-11T18:01:00+0200 IC 1632 to Biel/Bienne leaves Basel SBB at 2019-06-11T18:03:00+0200 IR 2335 to Erstfeld leaves Basel SBB at 2019-06-11T18:04:00+0200 S6 87852 to Basel Bad Bf leaves Basel SBB at 2019-06-11T18:04:00+0200

Creating an intent for querying connections

The screenshot shows the Dialogflow interface in Mozilla Firefox. On the left, the navigation sidebar includes 'Intents' (selected), 'Entities', 'Knowledge [beta]', 'Fulfillment', 'Integrations', 'Training', 'History', 'Analytics', 'Prebuilt Agents', 'Small Talk', and 'Docs'. At the bottom, it says 'Standard Free' and 'Upgrade'. The main area shows a intent named 'connection' with the following details:

- Action and parameters:** An action named 'connection' is defined with the following parameters:

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	from	@station_name	\$from	<input type="checkbox"/>
<input type="checkbox"/>	station_name1	@station_name	\$station_name1	<input type="checkbox"/> <input type="button" value="::"/>
<input type="checkbox"/>	date	@sys.date	\$date	<input type="checkbox"/>
<input type="checkbox"/>	time	@sys.time	\$time	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>
- Responses:** A 'Text response' is selected under the 'DEFAULT' tab.
- Agent Interaction:** A preview window shows a user message 'Show me trains leaving Basel tomorrow at 18:00.' and a response from the agent listing train departures.
- Intent Details:** The intent is named 'stationboard' with an 'ACTION' of 'Not available' and a parameter 'time' with a value of '2019-06-11T18:00:00+0200'.

List item:

- We have to change the parameter names: “station_name” to “from” and “station_name1” to “to”

Creating an intent for querying connections

The screenshot shows the Dialogflow console interface. On the left, the navigation sidebar includes sections for Intents, Entities, Knowledge [beta], Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard, and Free/Upgrade. The main workspace is titled "connection".

Action and parameters:

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	from	@station_name	\$from	<input type="checkbox"/>	Where are you I...
<input checked="" type="checkbox"/>	to	@station_name	\$to	<input type="checkbox"/>	What is your de...
<input type="checkbox"/>	date	@sys.date	\$date	<input type="checkbox"/>	—
<input type="checkbox"/>	time	@sys.time	\$time	<input type="checkbox"/>	—
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	—

Responses:

DEFAULT

Text response

Try it now:

See how it works in Google Assistant.

Agent:

USER SAYS

Show me trains leaving Basel tomorrow at 18:00.

COPY CURL

DEFAULT RESPONSE

S3 17365 to Olten leaves Basel SBB at 2019-06-11T18:01:00+0200 IC 1632 to Biel/Bienne leaves Basel SBB at 2019-06-11T18:03:00+0200 IR 2335 to Erstfeld leaves Basel SBB at 2019-06-11T18:04:00+0200 S6 87852 to Basel Bad Bf leaves Basel SBB at 2019-06-11T18:04:00+0200

INTENT: stationboard

ACTION: Not available

PARAMETER	VALUE
time	2019-06-11T18:00:00+02:00

Creating an intent for querying connections

The screenshot shows the Dialogflow API Client interface in Mozilla Firefox. The URL is <https://console.dialogflow.com/api-client/#/agent/6b9496d1-82f1-4d2f-9dc8-abcaae9addf1/newIntent>. The left sidebar shows navigation options like Intents, Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main area is titled "connection". It has input fields for "Enter name" (with "connection" selected), "Enter entity" (empty), and "Enter value" (empty). A "SAVE" button is visible. Below this is a "Responses" section with a "DEFAULT" tab. Under "Text response", there is a text input field with placeholder "Enter a text response". A "SET AS END OF CONVERSATION" toggle switch is present. To the right, a "Try it now" section shows a snippet of code: "See how it works in Google Assistant." Below this is an "Agent" section with a "USER SAYS" field containing "Show me trains leaving Basel tomorrow at 18:00." and a "COPY CURL" button. The "DEFAULT RESPONSE" dropdown shows a list of train departures from Basel. At the bottom, the "INTENT" is listed as "stationboard", "ACTION" as "Not available", and a parameter "time" with a value of "2019-06-11T18:04:00+0200".

Dialogflow – Mozilla Firefox

https://console.dialogflow.com/api-client/#/agent/6b9496d1-82f1-4d2f-9dc8-abcaae9addf1/newIntent

connection

Enter name: connection

Enter entity:

Enter value:

SAVE

Try it now

See how it works in Google Assistant.

Agent

USER SAYS

Show me trains leaving Basel tomorrow at 18:00.

COPY CURL

DEFAULT RESPONSE

S3 17365 to Olten leaves Basel SBB at 2019-06-11T18:01:00+0200 IC 1632 to Biel/Bienne leaves Basel SBB at 2019-06-11T18:03:00+0200 IR 2335 to Erstfeld leaves Basel SBB at 2019-06-11T18:04:00+0200 S6 87852 to Basel Bad Bf leaves Basel SBB at 2019-06-11T18:04:00+0200

INTENT

stationboard

ACTION

Not available

PARAMETER

time

VALUE

2019-06-11T18:04:00+0200

Test the new connection intent

Dialogflow - Mozilla Firefox
https://console.dialogflow.com/api-client/#/agent/6b9496d1-82f1-4d2f-9dc8-abcaae9addf1/editIntent/bcf2a2a0-1750-4eb2-9719-e84d316e7191/ 170%

The screenshot shows the Dialogflow interface in Mozilla Firefox. On the left, the navigation sidebar includes sections for transportbot, Intent (selected), Entities, Knowledge [beta], Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Standard Free, and Upgrade. The main workspace is titled 'connection'. It contains sections for Contexts, Events, and Training phrases. Under 'Training phrases', there are two entries: 'Show me connections from Zürich to Basel today at 12:00.' and 'Show me trains from Zürich to Basel today at 12:00.'. Below this is the 'Action and parameters' section, which lists three parameters: 'from' (REQUIRED, checked), 'to' (REQUIRED, checked), and 'date' (unchecked). The 'from' parameter is mapped to \$from with entities @station_name and prompts 'Where are you...' and 'What is your de...'. The 'to' parameter is mapped to \$to with entities @station_name and prompts 'Where are you...' and 'What is your de...'. The 'date' parameter is mapped to \$date with entities @sus.date and no prompts. To the right, the 'Agent' section shows a user message 'Show me connections from Zurich to Basel today at 12:00.' and a system response 'connection from Zürich HB to Basel SBB leaves at 2019-06-10T00:08:00+0200 on platform 14'. The 'INTENT' section shows 'connection', 'ACTION' shows 'Not available', and 'PARAMETER' shows 'from' with value 'Zürich HB' and 'time' with value '2019-06-11T 00:00:00+0200'.

Try it now

See how it works in Google Assistant.

Agent

USER SAYS COPY CURL

Show me connections from Zurich to Basel today at 12:00.

DEFAULT RESPONSE

connection from Zürich HB to Basel SBB leaves at 2019-06-10T00:08:00+0200 on platform 14

connection from Zürich HB to Basel SBB leaves at 2019-06-10T05:08:00+0200 on platform 15

connection from Zürich HB to Basel SBB leaves at 2019-06-10T06:00:00+0200 on platform 14

INTENT

connection

ACTION

Not available

PARAMETER VALUE

from Zürich HB

time 2019-06-11T 00:00:00+0200

Connect the Dialogflow Bot to a Webhook running on your local machine

- We provide code for a simple server that serves as a webhook for the bot we just created. You can run that server using the command “python -m main” (be sure that you have your anaconda environment activated)
- The server will run on port 5000 on your localhost.
- To make this server accessible for Dialogflow we will use “ngrok”. Ngrok creates an url like “<http://a178d27d.ngrok.io>” that will tunnel to your localhost.
- Download ngrok from: <https://ngrok.com/> and follow the installation instructions

Connect the Dialogflow Bot to a Webhook running on your local machine

- Run “ngrok http 5000” to create the tunnel.

```
pius@crunchy:~/work/transport_bot/transport_bot_v2
ngrok by @inconshreveable

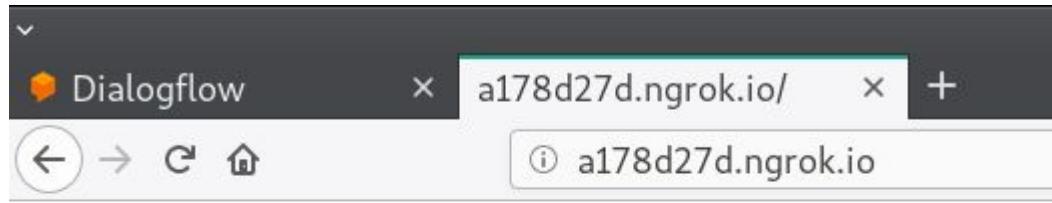
Session Status           online
Session Expires         6 hours, 57 minutes
Update                  update available (version 2.3.29, Ctrl-U to update)
Version                 2.3.29
Region                  United States (us)
Web Interface           http://127.0.0.1:4040
Forwarding              http://a178d27d.ngrok.io -> http://localhost:1818
Forwarding              https://a178d27d.ngrok.io -> http://localhost:1818

Connections             ttl     opn      rt1      rt5      p50      p90
                        5       0       0.00    0.00    0.29    0.42

HTTP Requests
-----
POST /webhook           200 OK
```

Connect the Dialogflow Bot to a Webhook running on your local machine

- You can test whether the tunnel works properly by browsing to the generated url.



Connect the Dialogflow Bot to a Webhook running on your local machine

The screenshot shows the Dialogflow Fulfillment settings page. On the left sidebar, 'Fulfillment' is selected. The main area shows the 'Webhook' configuration with the URL set to <https://5178d27d.ngrok.io/webhook>. The 'ENABLED' toggle switch is turned on. To the right, there's a 'Try it now' section with a text input field containing 'See how it works in Google Assistant.' Below it, an 'Agent' section shows a user message 'Show me connections from Zurich to Basel today at 12:00.' and a response 'connection from Zürich HB to Basel SBB leaves at 2019-06-10T00:08:00+0200 on platform 14 connection from Zürich HB to Basel SBB leaves at 2019-06-10T05:08:00+0200 on platform 15 connection from Zürich HB to Basel SBB leaves at 2019-06-10T06:00:00+0200 on platform 14'. At the bottom, there's an 'Inline Editor' section for Cloud Functions for Firebase.

- Change the fulfillment url to your ngrok url /webhook
- Test that everything is still working by entering a few example sentences in the text box on the top right.

Your turn: improve the bot!

- Look at the transport api documentation:
<https://transport.opendata.ch/docs.html> and figure out new intents or how to improve existing intents with the data provided.
- Bulk import a list of stations to extend the recognized station names.
- Give more example sentences for our intents so the bot recognizes our intents more consistently.
- Improve the responses.
- etc.